Uživatelská příručka Calibre

Vydání 2.46.0

Kovid Goyal

December 11, 2015

Obsah

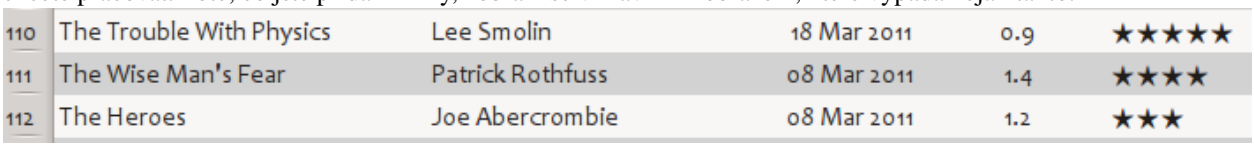
1 Oddíly **3**
1.1 Grafické uživatelské rozhraní . . . 3
1.2 Přidávání oblíbených webových stránek se zprávami . . . 23
1.3 Prohlížeč e-knih . . . 52
1.4 Převod e-knihy . . . 55
1.5 Úprava e-knih . . . 71
1.6 Porovnání e-knih . . . 100
1.7 Úprava metadat e-knihy . . . 102
1.8 Nejčastější dotazy . . . 105
1.9 Kurzy . . . 126
1.10 Přizpůsobení Calibre . . . 193
1.11 Rozhraní příkazového řádku . . . 228
1.12 Nastavení vývojového prostředí calibre . . . 260
1.13 Slovník pojmů . . . 282

Rejstřík modulů Pythonu **285**

Rejstřík **287**

Calibre je správce knihovny e-knih. Může zobrazovat, převádět a katalogizovat e-knihy ve většině hlavních formátů e-knih. Může také komunikovat s mnoha čtecími zařízeními e-knih. Může přejít na internet a načíst metadata pro vaše knihy. Může stahovat noviny a převést je na e-knihy pro pohodlné čtení. Je víceplatformové, funguje na systémech Windows, Linux a OSX.

Právě jste spustili Calibre. Co uděláte nyní? Než může Calibre udělat cokoliv s vašimi e-knihami, musí o nich nejdříve vědět. Přetáhněte pár souborů e-knih do Calibre nebo klikněte na tlačítko "Přidat knihy" a najděte knihy, se kterými chcete pracovat. Poté, co jste přidali knihy, zobrazí se v hlavním zobrazení, které vypadá nějak takto:


110 The Trouble With Physics Lee Smolin 18 Mar 2011 0.9 ★★★★★
111 The Wise Man's Fear Patrick Rothfuss 08 Mar 2011 1.4 ★★★★
112 The Heroes Joe Abercrombie 08 Mar 2011 1.2 ★★★

Poté, co jste spokojeně obdivovali seznam knih, které jste právě přidali, nejspíš si budete chtít nějakou přečíst. Abyste to mohli udělat, budete muset převést knihu do formátu, kterému vaše čtečka rozumí. Při prvním spuštění Calibre se spustí Uvítací průvodce a nastaví Calibre pro vaše čtecí zařízení. Převod je snadná věc. Stačí vybrat knihu, kterou chcete převést, a pak klikněte na tlačítko "Převést knihy". Prozatím ignorujte všechny volby a klikněte na "OK". V pravém dolním rohu se začne otáčet malá ikona. Jakmile se přestane otáčet, vaše převedená kniha je připravena. Klikněte na tlačítko "Zobrazit" pro čtení knihy.

Pokud chcete číst knihu ve své čtečce, připojte ji k počítači, počkejte, dokud ji Calibre nerozpozná (10-20 sekund), a pak klikněte na tlačítko "Odeslat do zařízení". Jakmile se ikona přestane znovu otáčet, odpojte svou čtečku a dejte se do čtení! Pokud jste v předchozím kroku nepřevedli knihu, Calibre ji převede automaticky do formátu, kterému vaše čtecí zařízení rozumí.

Abyste začali s pokročilejším používáním, měli byste si přečíst o *grafickém uživatelském rozhraní* (stránka 3). Pro ještě větší výkonnost a všestrannost si projděte `generated/en/cli-index`. Také shledáte užitečným seznam *nejčastějších dotazů* (stránka 105).

KAPITOLA 1

Oddíly

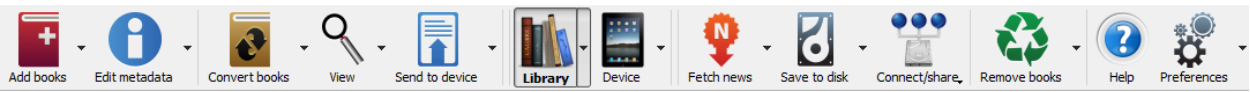
1.1 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní *(GUI)* poskytuje přístup k celé správě knihovny a funkcím převodů formátů e-knih. Základní pracovní postup pro používání Calibre je nejdříve přidat knihy do knihovny z vašeho pevného disku. Calibre se automaticky pokusí přečíst metadata z knih a přidat je do své vnitřní databáze. Jakmile jsou v databázi, můžete na nich provádět různé *Akce* (stránka 3), což zahrnuje převod z jednoho formátu do druhého, přenos do čtecího zařízení, zobrazení v počítači a úpravu metadat. Posledně zmíněné zahrnuje úpravu obálky, popisu a štítků společně s ostatními podrobnostmi. Pamatujte, že Calibre vytváří kopie souborů, které do něj přidáváte. Vaše původní soubory zůstanou nedotčené.

Rozhraní je rozděleno do různých oddílů:

- *Akce* (stránka 3)
- *Předvolby* (stránka 10)
- *Katalogy* (stránka 10)
- *Hledání a řazení* (stránka 11)
- *Rozhraní hledání* (stránka 11)
- *Ukládání hledání* (stránka 13)
- *Virtuální knihovny* (stránka 15)
- *Odhadování metadat z názvů souborů* (stránka 15)
- *Podrobnosti o knize* (stránka 16)
- *Prohlížeč štítků* (stránka 18)
- *Mřížka obálek* (stránka 20)
- *Prohlížeč obálek* (stránka 21)
- *Rychlé zobrazení* (stránka 21)
- *Úlohy* (stránka 22)
- *Klávesové zkratky* (stránka 22)

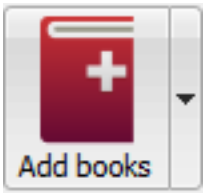
1.1.1 Akce



Panel nástrojů akcí poskytuje vhodné zkratky k běžně používaným akcím. Pokud kliknete pravým tlačítkem myši na tlačítka, můžete provést varianty výchozí akce. Pamatujte, prosím, že panel nástrojů akcí bude vypadat mírně odlišně v závislosti na tom, zda máte k počítači připojenou čtečku e-knih.

- *Přidat knihy* (stránka 4)
- *Upravit metadata* (stránka 5)
- *Převést knihy* (stránka 5)
- *Zobrazit* (stránka 6)
- *Odeslat do zařízení* (stránka 6)
- *Načíst zprávy* (stránka 7)
- *Knihovna* (stránka 7)
- *Zařízení* (stránka 8)
- *Uložit na disk* (stránka 8)
- *Připojit nebo sdílet* (stránka 9)
- *Odebrat knihy* (stránka 9)

Přidat knihy



Akce *Přidat knihy* má šedm variant přístupných kliknutím prvým tlačítkem myši na tlačítko.

1. **Přidat knihy z jednoho adresáře**: Otevře dialogové okno výběru souborů a umožní zadat, které knihy v adresáři by měly být přidány. Tato akce je *kontextově závislá*, t.j. že závisí na tom, který *katalog* (stránka 10) jste vybrali. Pokud jste vybrali *Knihovnu*, budou knihy přidány do knihovny. Pokud jste vybrali čtečku e-knih, budou knihy nahrány do zařízení, a tak dále.
2. **Přidat knih z adresářů včetně podadresářů (Jedna kniha na adresář, předpokládá, že každý soubor e-knihy je stejná kniha v jiném formátu)**: Umožňuje vám zvolit adresář. Adresář a všechny jeho podadresáře jsou prohledány rekurzivně a všechny nalezené e-knihy jsou přidány do knihovny. Calibre předpokládá, že každý adresář obsahuje jednu knihu. Všechny soubory e-knih v adresáři jsou považovány za stejnou knihu v jiných formátech. Tato akce je opak akce *Uložit na disk* (stránka 8), t.j. můžete *Uložit na disk*, odstranit knihy a znovu je přidat bez ztráty informací kromě data (toto předpokládá, že jste nezměnili žádné nastavení pro akci Uložit na disk).
3. **Přidat knihy z adresářů včetně podadresářů (Více knih na adresář, předpokládá, že každý soubor e-knihy je jiná kniha)**: Umožňuje vám zvolit adresář. Adresář a všechny jeho podadresáře jsou prohledány rekurzivně a všechny nalezené e-knihy jsou přidány do knihovny. Calibre předpokládá, že každý adresář obsahuje více knih. Všechny soubory e-knih se stejným názvem v adresáři jsou považovány za stejnou knihu v jiných formátech. E-knihy s jinými názvy jsou přidány jako jiné knihy.
4. **Přidat více knih z archivu (ZIP nebo RAR)**: Umožňuje vám přidat více e-knih, které jsou uloženy ve vybraných souborech ZIP nebo RAR. Je to vhodná zkratka, která zabraňuje tomu, abyste museli nejdříve rozbalit archiv, a potom přidat knihy prostřednictvím jedné ze dvou výše uvedených voleb.
5. **Přidat prázdnou knihu (Záznam knihy bez formátů)**: Umožňuje vám vytvořit prázdný záznam knihy. To může být použito pro následné ruční vyplnění informací o knize, kterou ještě nemusíte mít ve své sbírce.
6. **Přidat z ISBN**: Umožňuje vám přidat jednu nebo více e-knih zadáním jejich ISBN.
7. **Přidat soubory do vybraných záznamů knih**: Umožňuje vám přidat nebo aktualizovat soubory přidružené k existující knize ve vaší knihovně.

Akce *Přidat knihy* může číst metadata z široké nabídky formátů e-knih. Navíc se pokusí odhadnout metadat z názvu souboru. Podívejte se na oddíl *Odhadování metadat z názvů souborů* (stránka 15), abyste zjistili, jak toto konfigurovat.

Pro přidání dodatečného formátu pro existující knihu můžete udělat kteroukoliv ze tří věcí:

1. Přetáhnout soubor na panel podrobností knihy na pravé straně hlavního okna
2. Kliknout pravým tlačítkem myši na tlačítko Přidat knihy a zvolit *Přidat soubory k vybraným záznamům knih*.
3. Kliknout na červené tlačítko Přidat formát této knihy v pravé horní oblasti dialogového okna *Upravit metadata* otevřeného akcí *Upravit metadata* (stránka 5).

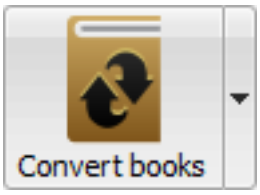
Upravit metadata

Akce *Upravit metadata* má čtyři varianty, které mohou být otevřeny kliknutím pravým tlačítkem myši na tlačítko.

1. **Upravit metadata jednotlivě**: Umožňuje vám upravit metadata knih postupně s možností načtení metadat včetně obálek z internetu. Také vám umožňuje přidat nebo odebrat konkrétní formáty e-knih z knihy.
2. **Upravit metadata hromadně**: Umožňuje vám upravit běžná pole metadat pro velké počty knih souběžně. Toto pracuje na všech knihách, které jste vybrali v *Zobrazení knihovny* (stránka 11).
3. **Stáhnout metadata a obálky**: Stáhne metadata a obálky (pokud jsou dostupné) pro knihy, které jste vybrali v seznamu knih.
4. **Sloučit záznamy knih**: Poskytne vám možnost sloučení metadat a formátů dvou nebo více záznamů knih. Můžete zvolit buď odstranit nebo ponechat záznamy, na které nebylo kliknuto nejdříve.

Pro více podrobností se podívejte na *Úprava metadat e-knihy* (stránka 102).

Převést knihy



E-knihy mohou být převedeny z několika formátů do jakéhokoliv formátu, který vaše čtečka e-knih upřednostňuje. Mnoho e-knih dostupných k zakoupení bude chráněno technologií Správa digitálních práv[1] *(DRM)*. Celibre tyto knihy nepřevede. Je snadné odebrat DRM z mnoha formátů, ale může to být nezákonné, budete si muset najít nástroje pro osvobození svých knih sami, a pak použít Calibre k jejich převodu.

Pro většinu lidí by převod měla být jednoduchá záležitost jednoho kliknutí. Pokud se chcete dozvědět více o procesu převodu, podívejte se na *Převod e-knihy* (stránka 55).

Akce *Převést knihy* má tři varianty přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Převést jednotlivě**: Umožňuje vám zadat volby převodu pro přizpůsobení převodu každé vybrané e-knihy.
2. **Hromadný převod**: Umožňuje vám zadat volby pouze jednou pro převod několika e-knih hromadně.
3. **Vytvořit katalog knih ve vaší knihovně Calibre**: Umožňuje vám generovat kompletní výpis knih ve vaší knihovně, včetně všech metadat, v několika formátech, jako jsou XML, CSV, BiBTeX, EPUB a MOBI. Katalog bude obsahovat všechny knihy aktuálně zobrazené v zobrazení knihovny. To vám umožní použít funkce hledání

[1] http://drmfree.calibre-ebook.com/about#drm

pro omezení knihy, které mají být uloženy do katalogu. Navíc pokud vyberete více knih pomocí myši, budou do katalogu přidány pouze tyto knihy. Pokud generujete katalog ve formátu e-knihy, jako je EPUB, MOBI nebo AZW3, příště když připojíte svou čtečku e-knih, bude katalog automaticky odeslán do zařízení. Pro více informací o tom, jak katalogy fungují, si přečtěte *Vytváření katalogů AZW3 • EPUB • MOBI* (stránka 187).

Zobrazit

Akce *Zobrazit* zobrazí knihu v programu prohlížeče e-knih. Calibre má vestavěný prohlížeč pro mnoho formátů e-knih. Pro ostatní formáty používá výchozí aplikace operačního systému. Které formáty by se měly otevírat vnitřním prohlížečem, můžete nastavit pomocí Předvolby->Chování. Pokud má kniha více než jeden formát, můžete zobrazit určitý formát kliknutím pravým tlačítkem myši na tlačítko.

Odeslat do zařízení

Akce *Odeslat do zařízení* má osm variant, které jsou přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Odeslat do hlavní paměti**: Vybrané knihy jsou přeneseny do hlavní paměti čtečky e-knih.
2. **Odeslat na paměťovou kartu A**: Vybrané knihy jsou přeneseny na paměťovou kartu A ve čtečce e-knih.
3. **Odeslat na paměťovou kartu B**: Vybrané knihy jsou přeneseny na paměťovou kartu B ve čtečce e-knih.
4. **Odeslat konkrétní formát do**: Vybrané knihy jsou přeneseny do vybraného úložiště v zařízení ve formátu, který zadáte.
5. **Vysunout zařízení**: Odpojí zařízení od Calibre.
6. **Nastavit výchozí akci odeslání do zařízení**: Umožňuje vám zadat, která z voleb, 1 až 5 nad nebo 7 pod, bude výchozí akcí, když kliknete na hlavní tlačítko.
7. **Odeslat a odstranit z knihovny**: Vybrané knihy jsou přeneseny do zvoleného úložiště v zařízení a pak **odstraněny** z knihovny.
8. **Načíst poznámky (experimentální)**: Přenese poznámky, které jste si udělali k e-knize ve vašem zařízení, k metadatům komentářů knihy v knihovně Calibre.

Název souboru a strukturu složek souborů odeslaných do zařízení můžete ovládat nastavením šablony v *Předvolby->Import a export->Odesílání knih do zařízení*. Také se podívejte na *The calibre template language* (stránka 133).

Načíst zprávy

Akce *Načíst zprávy* stahuje zprávy z různých webových stránek a převádí je na e-knihu, kterou můžete číst ve své čtečce e-knih. Obyčejně je nově vytvořená e-kniha přidána do vaší knihovny e-knih, ale pokud je v době dokončení stahování připojena čtečka e-knih, zprávy jsou také automaticky nahrány do čtečky.

Akce *Načíst zprávy* používá jednoduché předpisy (10-15 řádků kódu) pro každou stránku se zprávami. Abyste se naučili, jak vytvořit předpisy pro vlastní zdroje zpráv, podívejte se na *Přidávání oblíbených webových stránek se zprávami* (stránka 23).

Akce *Načíst zprávy* má tři varianty, přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Naplánovat stahování zpráv**: Umožňuje vám naplánovat stahování z vámi vybraných zdrojů zpráv ze seznamu stovek dostupných. Plánování může být nastaveno jednotlivě pro každý zdroj zpráv, který vyberete, a plánování je flexibilní, což vám umožňuje vybrat konkrétní dny v týdnu nebo počet dnů mezi stahováním.
2. **Přidat vlastní zdroj zpráv**: Umožňuje vám vytvořit jednoduchý předpis pro stahování zpráv z vlastní stránky se zprávami, ke kterým chcete mít přístup. Vytváření předpisu může být tak jednoduché, jako zadání adresy URL informačního kanálu RSS, nebo můžete být více předpisující vytvořením kódu založeném na Pythonu pro tuto úlohu. Více informací naleznete na *Přidávání oblíbených webových stránek se zprávami* (stránka 23).
3. **Stáhnout všechny naplánované zdroje zpráv**: Způsobí, že Calibre okamžitě začne stahovat všechny zdroje zpráv, které jste naplánovali.

Knihovna

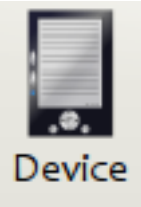
Akce *Library* vám umožňuje vytvářet, přepínat mezi, přejmenovávat nebo odebírat knihovny. Calibre vám umožňuje vytvořit tolik knihoven, kolik budete chtít. Mohli byste například vytvořit knihovnu pro sci-fi, knihovnu pro naučnou literaturu, knihovnu pro cizí jazyk, knihovnu pro projekt nebo jakoukoliv strukturu, která vyhovuje vašim potřebám. Knihovny jsou nejvyšší organizační struktura v Calibre. Každá knihovna má svou vlastní sadu knih, štítků, kategorií a základní umístění úložiště.

1. **Přepnout nebo vytvořit knihovnu... **: Umožňuje vám a) připojit se k již existující knihovně Calibre v jiném umístění, b) vytvořit prázdnou knihovnu v novém umístění nebo c) přesunout aktuální knihovnu do nově zadaného umístění.
2. **Rychlé přepnutí**: Umožňuje vám přepínat mezi knihovnami, které byly registrovány nebo vytvořeny v Calibre.
3. **Přejmenovat knihovnu**: Umožňuje vám přejmenovat knihovnu.
4. **Odstranit knihovnu**: Umožňuje vám odregistrovat knihovnu z Calibre.
5. **<název knihovny>**: Akce 5, 6 atd... vám poskytuje okamžitý přístup k přepnutí mezi více knihovnami, které jste vytvořili nebo připojili. Tento seznam obsahuje pouze 5 nejčastěji používaných knihoven. Pro kompletní seznam použijte nabídku Rychlé přepnutí.
6. **Údržba knihovny**: Umožňuje vám zkontrolovat v aktuální knihovně problémy s konzistencí dat a obnovit databázi aktuální knihovny ze záloh.

Poznámka: Metadata o vašich e-knihách, např. název, autor a štítky, jsou uloženy v jednom souboru ve složce knihovny Calibre nazvané metadata.db. Pokud se tento soubor poškodí (velice vzácná událost), můžete ztratit metadata. Naštěstí Calibre automaticky zálohuje metadata pro každou jednotlivou knihu ve složce knihy jako soubor OPF. Použitím akce 'Obnovit databázi' pod výše popsanou 'Údržbou knihovny' pro vás může Calibre znovu sestavit soubor metadata.db z jednotlivých souborů OPF.

Můžete kopírovat nebo přesouvat knihy mezi různými knihovnami (jakmile budete mít nastavenou více než jednu knihovnu) kliknutím pravým tlačítkem myši na knihu a výběrem akce *Kopírovat do knihovny*.

Zařízení



Akce *Zařízení* vám umožňuje zobrazit knihy v hlavní paměti nebo na paměťových kartách vašeho zařízení, nebo vysunout zařízení (odpojit ho od Calibre). Tato ikona se automaticky zobrazí na hlavním panelu nástrojů Calibre, když připojíte podporované zařízení. Můžete na ni kliknout, abyste viděli knihy ve svém zařízení. Můžete také přetáhnout knihy ze své knihovny Calibre na ikonu, abyste je přenesli do svého zařízení. A naopak můžete přetáhnout knihy ze svého zařízení na ikonu knihovny na panelu nástrojů, abyste přenesli knihy ze svého zařízení do knihovny Calibre.

Uložit na disk

Akce *Uložit na disk* má pět variant, které jsou přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Uložit na disk**: Uloží vybrané knihy na disk roztříděné do adresářů. Struktura adresářů vypadá takto:

```
Author_(sort)
    Title
        Book Files
```

Název souboru a strukturu složek souborů uložených na disk můžete ovládat nastavením šablony v *Předvolby->Import a export->Ukládání knih na disk*. Také se podívejte na *The calibre template language* (stránka 133).

2. **Uložit na disk do jednoho adresáře**: Uloží vybrané knihy na disk do jednoho adresáře.

Pro 1. a 2. jsou všechny dostupné formáty, stejně jako metadata, uloženy na disk pro každou vybranou knihu. Metadata jsou uložena do souboru OPF. Uložené knihy mohou být znovu importovány do knihovny bez jakékoliv ztráty informací pomocí akce *Přidat knihy* (stránka 4).

3. **Uložit na disk pouze <vámi upřednostňovaný> formát**: Uloží vybrané knihy na disk v adresářové struktuře, jak je zobrazeno v (1.), ale pouze ve vámi upřednostňovaném formátu e-knihy. Svůj upřednostňovaný formát můžete nastavit v *Předvolby->Chování->Upřednostňovaný výstupní formát*

4. **Uložit na disk do jednoho adresáře pouze <vámi upřednostňovaný> formát**: Uloží vybrané knihy na disk do jednoho adresáře, ale pouze ve vámi upřednostňovaném formátu e-knihy. Svůj upřednostňovaný formát můžete nastavit v *Předvolby->Chování->Upřednostňovaný výstupní formát*
5. **Uložit na disk jeden formát...**: Uloží vybrané knihy na disk v adresářové struktuře, jak je zobrazeno v (1.), ale pouze ve formátu, který jste vybrali z rozbalovacího seznamu.

Připojit nebo sdílet

Akce *Připojit nebo sdílet* vám umožňuje ručně se připojit k zařízení nebo složce ve vašem počítači. Také vám to umožňuje nastavit knihovnu Calibre pro přístup prostřednictvím webového prohlížeče nebo e-mailu.

Akce *Připojit nebo sdílet* má čtyři varianty, přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Připojit ke složce**: Umožňuje vám připojit se ke kterékoliv složce ve vašem počítači, jako by to bylo zařízení a využívat s touto složkou všechno vybavení, které má Calibre pro zařízení. To je užitečné, pokud vaše zařízení nemůže být podporováno v Calibre, ale je dostupné jako disk USB.
2. **Připojit k iTunes**: Umožňuje vám připojit se k vaší databázi knih iTunes, jako by to bylo zařízení. Jakmile jsou knihy odeslány do iTunes, můžete použít iTunes, aby byly dostupné pro vaše různá iZařízení.
3. **Spustit server s obsahem**: Spustí vestavěný webový server Calibre. Když je spuštěný, bude vaše knihovna Calibre přístupná prostřednictvím webového prohlížeče z internetu (pokud si to zvolíte). Jak je přistupováno k webovému serveru můžete konfigurovat nastavením předvoleb v *Předvolby->Sdílení->Sdílení po síti*
4. **Nastavit sdílení knih na základě e-mailu**: Umožňuje sdílení knih a kanálů zpráv e-mailem. Po nastavení e-mailové adresy pro tuto volbu bude Calibre odesílat aktualizace zpráv a knih na zadané e-mailové adresy. Jak Calibre odesílá e-maily můžete konfigurovat nastavením předvoleb v *Předvolby->Sdílení->Sdílení knih e-mailem*. Jakmile nastavíte jednu nebo více e-mailových adres, bude tato položka nabídky nahrazena položkami nabídky pro odesílání knih na nastavené e-mailové adresy.

Odebrat knihy

Akce *Odebrat knihy* **trvale odstraní knihy**, takže ji používejte opatrně. Je *kontextově závislá*, t.j. závisí na tom, který *katalog* (stránka 10) jste vybrali. Pokud jste vybrali *Knihovnu*, knihy budou odebrány z knihovny. Pokud jste zvolili zařízení čtečky e-knih, budou knihy odebrány ze zařízení. Abyste odebrali pouze určitý formát pro danou knihu, použijte akci *Upravit metadata* (stránka 5). Odebrání knih má také pět variant, ke kterým můžete získat přístup kliknutím pravým tlačítkem myši na tlačítko.

1. **Odebrat vybrané knihy**: Umožňuje vám **trvale** odebrat všechny knihy, které jsou vybrány v seznamu knih.

2. **Odebrat soubory konkrétního formátu z vybraných knih**: Umožňuje vám **trvale** odebrat soubory e-knih konkrétního formátu z knih, které jsou vybrány v seznamu knih.
3. **Odebrat všechny formáty z vybraných knih, kromě...**: Umožňuje vám **trvale** odebrat soubory e-knih všech formátů kromě zadaného formátu z knih, které jsou vybrány v seznamu knih.
4. **Odebrat všechny formáty z vybraných knih**: Umožňuje vám **trvale** odebrat všechny soubory e-knih z knih, které jsou vybrány v seznamu knih. Zůstanou pouze metadata.
5. **Odebrat obálky z vybraných knih**: Umožňuje vám **trvale** odebrat všechny soubory obrázků obálky z knih, které jsou vybrány v seznamu knih.
6. **Odebrat odpovídající knihy ze zařízení**: Umožňuje vám odebrat soubory e-knih z připojeného zařízení, které odpovídají knihám, které jsou vybrány v seznamu knih.

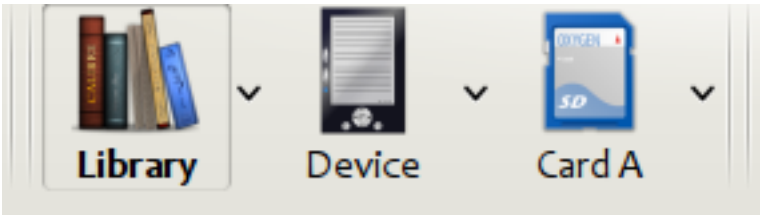
Poznámka: Pamatujte, že když použijete Odebrat knihy k odstranění knih z vaší knihovny Calibre, je záznam knihy trvale odstraněn, ale na Windows a OS X jsou soubory umístěny do koše. To vám umožňuje obnovit je, pokud si to rozmyslíte.

1.1.2 Předvolby

Akce *Předvolby* vám umožňuje změnit způsob, jakým různé aspekty Calibre fungují. Má čtyři varianty přístupné kliknutím pravým tlačítkem myši na tlačítko.

1. **Předvolby**: Umožňuje vám změnit způsob, jakým různé aspekty Calibre fungují. Kliknutí na tlačítko také provede tuto akci.
2. **Spustit uvítacího průvodce**: Umožňuje vám spustit Uvítacího průvodce, který se objevil, když jste poprvé spustili Calibre.
3. **Získat moduly pro rozšíření Calibre**: Otevře nové okno, které zobrazí moduly pro Calibre. Tyto moduly jsou vyvíjeny třetími stranami pro rozšíření funkčnosti Calibre.
4. **Restartovat v režimu ladění**: Umožňuje vám povolit režim ladění, který může pomáhat vývojářům Calibre při řešení problémů, na které narazíte v programu. Pro většinu uživatelů by toto mělo zůstat zakázáno, pokud jim vývojáři neudělí pokyn, aby to povolili.

1.1.3 Katalogy



Katalog je kolekce knih. Calibre může spravovat dva typy různých katalogů:

1. **Knihovna**: Toto je kolekce knih uložených ve vaší knihovně Calibre ve vašem počítači.
2. **Zařízení**: Toto je kolekce knih uložených ve vaší čtečce e-knih. Bude dostupná, když připojíte čtečku k počítači.

Mnoho operací, jako je přidávání knih, odstraňování, zobrazování atd., je kontextově závislých. Takže například pokud kliknete na tlačítko Zobrazit, když máte vybraný katalog **Zařízení**, Calibre otevře pro zobrazení soubory v zařízení. Pokud máte vybraný katalog **Knihovna**, budou namísto toho otevřeny soubory ve vaší knihovně Calibre.

1.1.4 Hledání a řazení

	Title	Author(s)	Size (MB)	Date	Rating	Publisher	Tags	Series
1	The Complete Works of William Shakespeare	William Shakespeare	2.4	02 Jan 2007	★★★★★	manybooks.net		
2	Stalky and Co.	Rudyard Kipling	0.2	19 Jan 2007	★★★★★	manybooks.net		
3	The Comedies of William Shakespeare	William Shakespeare	2.1	15 Mar 2007	★★★★★			
4	The Histories of William Shakespeare	William Shakespeare	1.5	15 Mar 2007	★★★★★		england, historical fiction	
5	The Tragedies of William Shakespeare	William Shakespeare	1.6	15 Mar 2007	★★★★★			
6	War and Peace	Leo Tolstoy	3.1	22 Aug 2007	★★★★★	gutenberg.org	classic	
7	Anna Karenina	Leo Tolstoy	1.9	22 Aug 2007	★★★★★	gutenberg.org	classic	
8	Guns, germs, and steel: the fates of human societies	Jared Diamond	0.4	29 Nov 2007	★★★★★	New York : W.W. Norton, c1997.		
9	A Game of Thrones	George R. R. Martin	1.3	23 Jan 2007	★★★★		fantasy	
10	A Clash of Kings	George R. R. Martin	1.4	25 Jan 2007	★★★★		fantasy	
11	A Storm of Swords	George R. R. Martin	1.9	27 Jan 2007	★★★★		fantasy	
12	A Feast for Crows	George R. R. Martin	1.7	29 Jan 2007	★★★★		fantasy	Song of Ice and Fire [4]

Oddíl Hledání a řazení vám umožňuje provést několik výkonných akcí na vaší kolekci knih.

- Můžete je seřadit podle názvu, autora, data, hodnocení atd. kliknutím na názvy sloupců. Můžete také "podřazovat", t.j. řadit podle více sloupců. Například pokud kliknete na sloupec název a pak na sloupec autor, knihy budou seřazeny podle autora a všechny položky pro stejného autora budou seřazeny podle názvu.
- Můžete hledat konkrétní knihu nebo sadu knih pomocí panelu hledání. Více o tom níže.
- Můžete rychle a pohodlně upravovat metadata dvojitým kliknutím na položku, kterou chcete změnit v seznamu.
- Můžete provádět *Akce* (stránka 3) na sadách knih. Pro výběr více knih můžete:
  - Ponechat stisknutou klávesu `Ctrl` a kliknout na knihy, které chcete vybrat.
  - Ponechat stisknutou klávesu `Shift` a kliknout na první a poslední knihu z rozsahu knih, které chcete vybrat.
- Můžete konfigurovat, která pole chcete zobrazit pomocí dialogového okna *Předvolby* (stránka 10).

1.1.5 Rozhraní hledání

Můžete prohledávat všechna metadata zadáním hledaných výrazů do panelu hledání. Vyhledávání nerozlišuje malá a velká písmena. Například:

```
Asimov Foundation format:lrf
```

Toto najde všechny knihy ve vaší knihovně, které mají ve svých metadatech `Asimov` a `Foundation` a jsou dostupné ve formátu LRF. Některé další příklady:

```
author:Asimov and not series:Foundation
title:"The Ring" or "This book is about a ring"
format:epub publisher:feedbooks.com
```

Vyhledávání je ve výchozím nastavení ‘obsahuje’. Položka odpovídá, pokud se hledaný řetězec objeví kdekoliv v označených metadatech. Jsou dostupné další dva druhy vyhledávání: vyhledávání rovnosti a vyhledávání pomocí regulárních výrazů <http://en.wikipedia.org/wiki/Regular_expression>‘_.

Vyhledávání rovnosti jsou označena přidáním předpony symbolu rovnosti (=) hledanému řetězci. Například dotazu `tag:=science"` bude odpovídat “science”, ale ne “science fiction” nebo “hard science”. Vyhledávání pomocí regulárních výrazů jsou označena přidáním předpony vlnovky (~) hledanému řetězci. Mohou být použity jakékoliv regulární výrazy kompatibilní s Pythonem[2] . Pamatujte, že zpětná lomítka používaná pro uvození speciálních znaků v regulárních výrazech musí být dvojitá, protože jednoduchá zpětná lomítka budou odebrána během analýzy dotazu. Například pro porovnání prostých závorek musíte zadat `\\(`. Vyhledávání pomocí regulárních výrazů jsou vyhledávání typu ‘obsahuje’, pokud výraz neobsahuje kotvy.

Pokud potřebujete hledat řetězec s rovnítkem nebo vlnovkou na začátku, přidejte před řetězec zpětné lomítko.

Uzavřete hledané řetězce uvozovkami (”), pokud řetězec obsahuje závorky nebo mezery. Například pro hledání štítku‘‘Science Fiction‘‘ byste museli hledat `tag:=science fiction"`. Pokud hledáte `tag:=science fiction` najdete všechny knihy se štítkem ‘science’ a obsahující slovo ‘fiction’ v kterýchkoliv metadatech.

Vytvořit pokročilé vyhledávací dotazy můžete snadno pomocí dialogového okna *Pokročilé hledání*, které otevřete kliknutím na tlačítko .

Pole dostupná pro vyhledávání jsou: `tag`, `title`, `author`, `publisher`, `series`, `series_index`, `rating`, `cover`, `comments`, `format`, `identifiers`, `date`, `pubdate`, `search`, `size` a vlastní sloupce. Pokud je připojeno zařízení, bude při hledání v zobrazení knihovny Calibre dostupné pole `ondevice`. Abyste našli název hledání (ve skutečnosti nazvaný *název vyhledávání*) pro vlastní sloupec, přejeďte myší nad záhlavím sloupce v zobrazení knihovny.

Syntaxe pro hledání dat je:

```
pubdate:>2000-1 Will find all books published after Jan, 2000
date:<=2000-1-3 Will find all books added to calibre before 3 Jan, 2000
pubdate:=2009 Will find all books published in 2009
```

Pokud je datum dvojznačné, je pro porovnání data použito aktuální národní prostředí. Například v národním prostředí mm/dd/yyy je 2/1/2009 interpretováno jako 1 úno 2009. V národním prostředí dd/mm/yyyy je interpretováno jako 2 led 2009. Jsou dostupné některé speciální řetězce dat. Řetězec `today` se převede na dnešní datum, ať je jakékoliv. Řetězce `yesterday` a `thismonth` (nebo přeložené ekvivalenty v aktuálním jazyce), také fungují. Navíc řetězec `daysago` (také přeložený) může být použit pro porovnání s datem před určitým počtem dnů. Například:

```
date:>10daysago
date:<=45daysago
```

Aby se zabránilo případným problémům s přeloženými řetězci při používání neanglické verze Calibre, jsou vždy dostupné řetězce `_today`, `_yesterday`, `_thismonth` a `_daysago`. Ty nejsou přeložené.

Takto můžete hledat knihy, které mají formát o určité velikosti:

```
size:>1.1M Will find books with a format larger than 1.1MB
size:<=1K  Will find books with a format smaller than 1KB
```

Data a číselná pole podporují relační operátory = (rovná se), > (větší než), >= (větší než nebo rovno), < (menší než), <= (menší než nebo rovno) a != (není rovno). Pole hodnocení jsou považována za číselná. Například hledání `rating:>=3` nalezne všechny knihy s hodnocením 3 nebo vyšším.

Můžete hledat počet položek v polích s více hodnotami, jako jsou štítky. Tato hledání začínají znakem `#`, pak použijte stejnou syntaxi jako číselná pole. Například abyste našli všechny knihy s více než 4 štítky, použijte `tags:#>4`. Abyste našli všechny knihy s přesně 10 štítky, použijte `tags:#=10`.

[2] https://docs.python.org/2/library/re.html

Indexy sérií jsou vyhledatelné. Pro standardní série je název hledání 'series_index'. Pro vlastní sloupce sérií použijte název hledání sloupce následovaný _index. Například abyste hledali indexy pro vlastní sloupec sérií pojmenovaný `#moje_serie`, použili byste název hledání `#moje_serie_index`. Indexy sérií jsou čísla, takže můžete použít relační operátory popsané výše.

Pro uložená hledání je použito speciální pole `search`. Takže pokud uložíte hledání s názvem "Knihy mé manželky", můžete do panelu hledání zadat `search:"Knihy mé manželky"` pro znovu použití uloženého hledání. Více o ukládání hledání níže.

Nepřítomnost nebo přítomnost pole můžete hledat pomocí speciálních hodnot "true" a "false". Například:

```
cover:false will give you all books without a cover
series:true will give you all books that belong to a series
comments:false will give you all books with an empty comment
format:false will give you all books with no actual files (empty records)
```

Sloupce Ano/Ne jsou vyhledatelné. Hledání `false`, `empty` nebo `blank` nalezne všechny knihy s nedefinovanými hodnotami ve sloupci. Hledání `true` nalezne všechny knihy, které nemají nedefinované hodnoty ve sloupci. Hledání `yes` nebo `checked` nalezne všechny knihy s `Ano` ve sloupci. Hledání `no` nebo `unchecked` nalezne všechny knihy s `Ne` ve sloupci. Pamatujte, že slova `yes`, `no`, `blank`, `empty`, `checked` a `unchecked` jsou přeložena; můžete použít buď ekvivalent aktuálního jazyka, nebo anglické slovo. Slova `true` a `false` a speciální hodnoty `_yes`, `_no` a `_empty` nejsou přeloženy.

Hierarchické položky (např. A.B.C) používají rozšířenou syntaxi pro porovnání počátečních částí hierarchie. To se provádí přidáním tečky mezi indikátor přesné shody (=) a text. Například dotaz `tags:=.A` nalezne štítky *A* a *A.B*, ale nenalezne štítky *AA* nebo *AA.B*. Dotaz `tags:=.A.B` nalezne štítky *A.B* a *A.B.C*, ale ne štítek *A*.

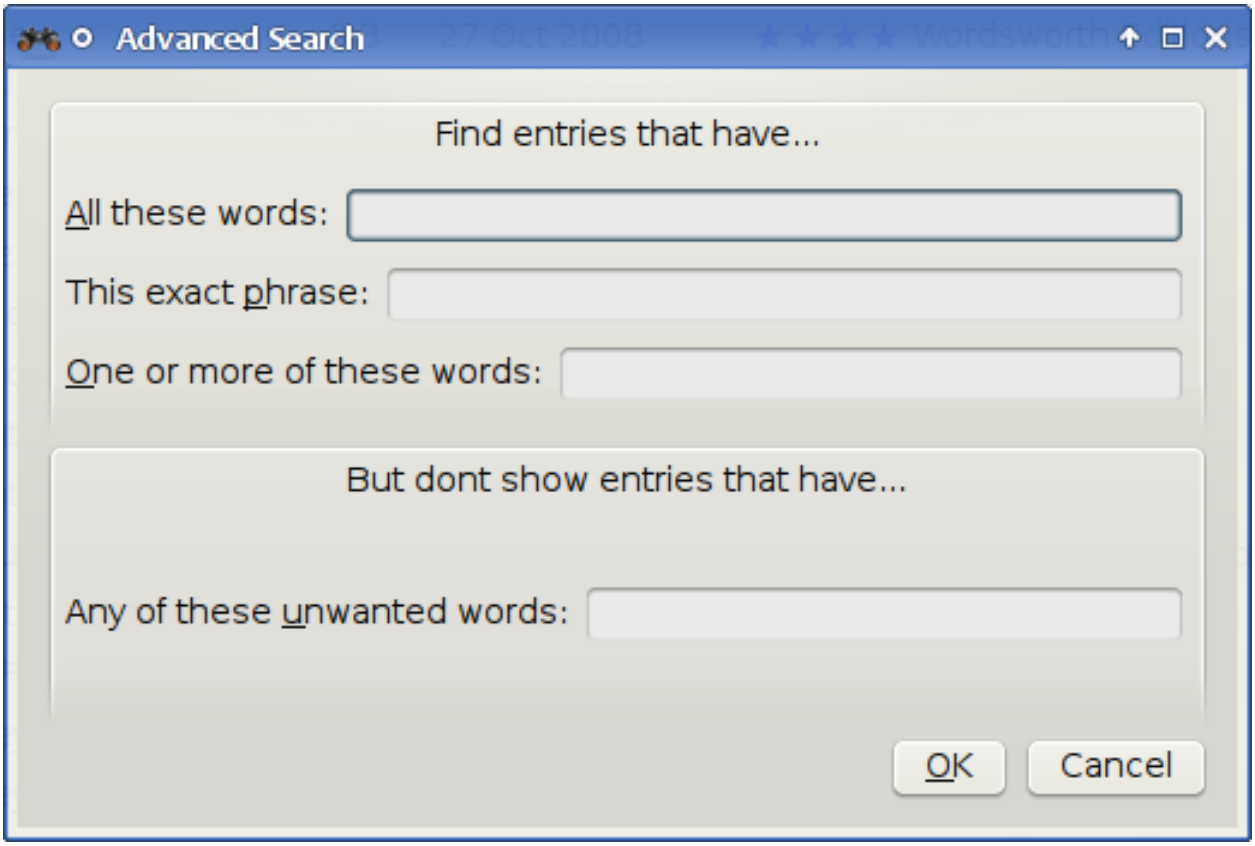
Identifikátory (např. isbn, doi, lccn atd.) také používají rozšířenou syntaxi. Za prvé pamatujte, že identifikátor má podobu `type:value`, jako u `isbn:123456789`. Rozšířená syntaxe umožňuje zadat nezávisle na sobě, jaký typ a hodnotu hledat. Části dotazu typ a hodnota mohou použít porovnání *rovnost*, *obsahuje* nebo *regulární výraz*. Příklady:

- `identifiers:true` najde knihy s jakýmkoliv identifikátorem.
- `identifiers:false` najde knihy bez identifikátoru.
- `identifiers:123` bude hledat knihy s jakýmkoliv typem majícím hodnotu obsahující *123*.
- `identifiers:=123456789` bude hledat knihy s jakýmkoliv typem majícím hodnotu rovnající se *123456789*.
- `identifiers:=isbn:` a `identifiers:isbn:true` najde knihy s typem rovnajícím se isbn majícím jakoukoliv hodnotu.
- `identifiers:=isbn:false` najde knihy s žádným typem rovnajícím se isbn.
- `identifiers:=isbn:123` najde knihy s typem rovnajícím se isbn majícím hodnotu obsahující *123*.
- `identifiers:=isbn:=123456789` najde knihy s typem rovnajícím se isbn majícím hodnotu rovnající se *123456789*.
- `identifiers:i:1` najde knihy s typem obsahujícím *i* majícím hodnotu obsahující *1*.

1.1.6 Ukládání hledání

Calibre vám umožňuje uložit často používané hledání pod speciálním názvem a pak znovu použít toto hledání pomocí jediného kliknutí. Abyste to udělali, vytvořte hledání buď zadáním do panelu hledání, nebo pomocí Prohlížeče štítků. Potom zadejte název, který chcete dát hledání, do pole Uložená hledání vedle panelu hledání. Klikněte na ikonu plus vedle pole uložených hledání pro uložení hledání.

Nyní můžete přistupovat ke svému uloženému hledání v Prohlížeči štítků pod "Hledání". Jedno kliknutím vám umožní snadno znovu použít jakékoliv libovolně složité hledání, aniž byste ho museli znovu vytvářet.



Obr. 1.1: *Dialogové okno Pokročilé hledání*

1.1.7 Virtuální knihovny

Virtuální knihovna je způsob, jak předstírat, že vaše knihovna Calibre má pouze několik knih, namísto celé její kolekce. Je to vynikající způsob, jak rozdělit velkou kolekci knih na menší, snadno ovladatelné bloky. Abyste zjistili, jak vytvořit a používat virtuální knihovny, podívejte se na kurz: *Virtuální knihovny* (stránka 191).

1.1.8 Odhadování metadat z názvů souborů

Obyčejně Calibre čte metadata zevnitř souboru knihy. Může však být nakonfigurováno tak, aby nemísto toho četlo metadata z názvu souboru, a to pomocí *Předvolby->Přidávání knih->Číst metadata z obsahu souboru*.

Můžete také určit, jak jsou čtena metadata z názvu souboru pomocí regulárních výrazů (viz Vše o používání regulárních výrazů v Calibre (stránka 163)). V oddílu konfiguračního dialogového okna *Přidávání knih* můžete zadat regulární výraz, který Calibre použije, aby se pokusilo odhadnout metadata z názvů souborů e-knih, které přidáte do knihovny. Výchozí regulární výraz je:

```
title - author
```

to znamená, že předpokládá, že všechny znaky až do první – jsou název knihy a následující znaky jsou autor knihy. Například název souboru:

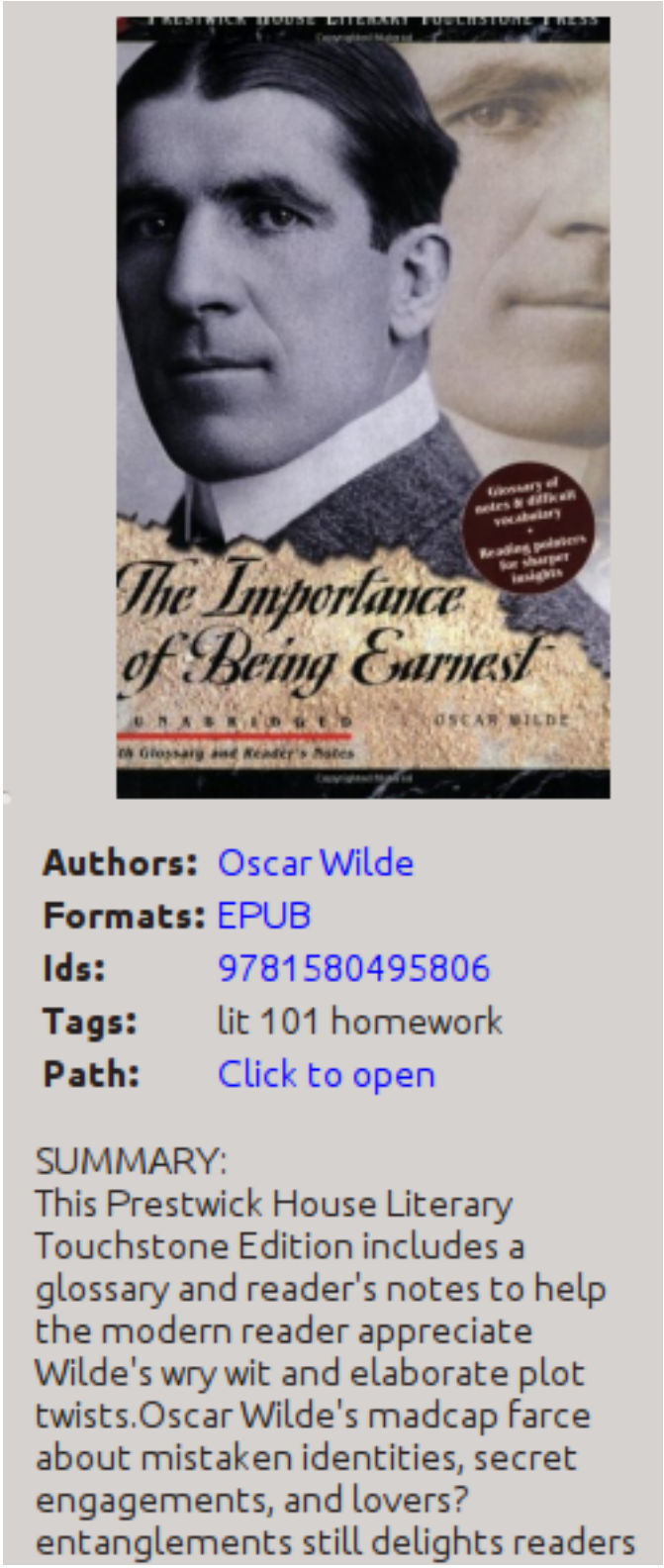
```
Foundation and Earth - Isaac Asimov.txt
```

bude interpretován, že má název: Foundation and Earth a autora: Isaac Asimov

---

Tip: Pokud název souboru neobsahuje spojovník, výše uvedený regulární výraz selže.

1.1.9 Podrobnosti o knize


The Importance of Being Earnest
OSCAR WILDE
Authors: Oscar Wilde
Formats: EPUB
Ids: 9781580495806
Tags: lit 101 homework
Path: Click to open
SUMMARY:
This Prestwick House Literary Touchstone Edition includes a glossary and reader's notes to help the modern reader appreciate Wilde's wry wit and elaborate plot twists.Oscar Wilde's madcap farce about mistaken identities, secret engagements, and lovers? entanglements still delights readers

Podrobnosti o knize zobrazují obálku a všechna metadata pro aktuálně vybranou knihu. Mohou být skryty pomocí

tlačítka v pravém dolním rohu hlavního okna Calibre. Na jména autorů zobrazená v panelu Podrobností o knize lze kliknout, ve výchozím nastavení vás přenesou na stránku autora na Wikipedii. Toto může být upraveno kliknutím pravým tlačítkem myši na jméno autora a výběrem Spravovat tohoto autora.

Podobně pokud stáhnete metadata pro knihu, panel Podrobností o knize vám automaticky zobrazí odkazy směřující na webové stránky pro knihu na Amazonu, WorldCatu atd., odkud byla metadata stažena.

Můžete kliknout pravým tlačítkem myši na jednotlivé formáty e-knih v panelu Podrobností o knize k jejich odstranění, porovnání s jejich původními verzemi, uložení na disk, otevření v externím programu atd.

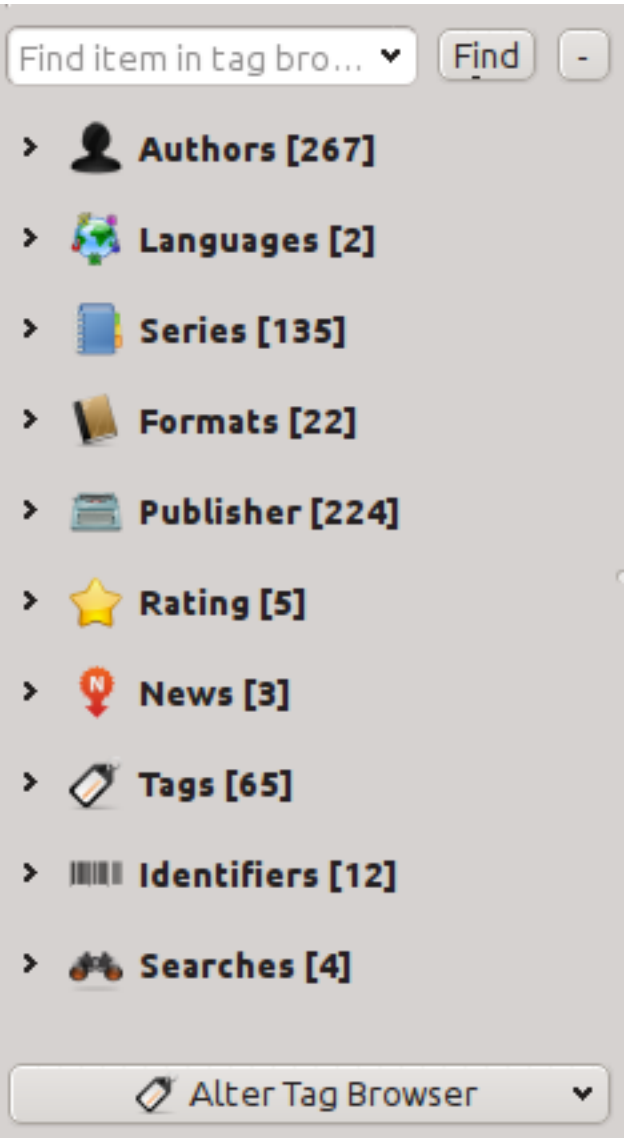
Obálku knihy můžete změnit jednoduše přetažením obrázku na panel podrobností o knize. Pokud si přejete upravit obrázek obálky v externím programu, jednoduše na něj klikněte pravým tlačítkem myši a zvolte *Otevřít v programu*.

Můžete také přidat soubory e-knihy do aktuální knihy přetažením souborů na panel Podrobností o knize.

Dvojité kliknutí na panel Podrobností o knize ji otevře v samostatném automaticky otevřeném okně.

A nakonec můžete přesně přizpůsobit, jaké informace jsou zobrazeny v panelu Podrobností o knize pomocí *Předvolby->Vzhled->Podrobnosti o knize*.

1.1.10 Prohlížeč štítků



Prohlížeč štítků vám umožňuje snadno procházet vaši kolekci podle autora, štítků, série atd. Pokud kliknete na kteroukoliv položku v Prohlížeči štítků, například jméno autora Isaac Asimov, pak je seznam knih na pravé straně je omezen, aby zobrazoval knihy od tohoto autora. Můžete kliknout rovněž na názvy kategorií. Například kliknutí na "Série" zobrazí všechny knihy v kterékoliv sérii.

První kliknutí na položku omezí seznam knih na ty, které obsahují nebo odpovídají položce. Jako pokračování výše uvedeného příkladu kliknutí na Isaaca Asimova zobrazí knihy od tohoto autora. Nové kliknutí na položku změní to, co je zobrazeno, v závislosti na tom, zda má položka podřízené položky (viz podkategorie a hierarchické položky níže). Jako pokračování příkladu s Isaacem Asimovem nové kliknutí na Isaaca Asimova omezí seznam knih na ty, které nejsou od Isaaca Asimova. Třetí kliknutí odebere omezení a zobrazí všechny knihy. Pokud podržíte klávesu Ctrl nebo Shift a kliknete na více položek, pak jsou vytvořena omezení na základě více položek. Například byste mohli držet klávesu Ctrl a kliknout na štítky Historická a Evropa pro nalezení knih o dějinách Evropy. Prohlížeč štítků funguje sestavováním vyhledávacích výrazů, které jsou automaticky zadávány do panelu hledání. Dívání se na to, co Prohlížeč štítků generuje, je dobrý způsob zjištění, jak vytvořit základní vyhledávací výrazy.

Položky v Prohlížeči štítků mají částečně barevné ikony. Množství barev závisí na průměrném hodnocení knih v dané

kategorii. Takže například pokud knihy od Isaaca Asimova mají průměrně čtyři hvězdičky, bude ikona pro Isaaca Asimova v Prohlížeči štítků barevná ze 4/5. Můžete přejet myší nad ikonou, abyste viděli průměrné hodnocení.

Položky na vnější úrovni v Prohlížeči štítků, jako jsou Autoři a Série, se nazývají kategorie. Můžete si vytvořit své vlastní kategorie, nazvané Uživatelské kategorie, které jsou užitečné pro organizování položek. Například můžete použít Editor uživatelských kategorií (klikněte na *Upravit Prohlížeč štítků->Spravovat autory, štítky atd.->Spravovat uživatelské kategorie*) pro vytvoření uživatelské kategorie nazvané Olíbení autoři, a pak umístit své oblíbené položky do kategorie. Uživatelské kategorie mohou mít podkategorie. Například uživatelská kategorie Oblíbení.Autoři je podkategorie Oblíbené. Možná také máte Oblíbené.Série, v takovém případě budou pod Oblíbené dvě podkategorie. Podkategorie mohou být vytvořeny kliknutím pravým tlačítkem myši na uživatelskou kategorii, výběrem "Přidat podkategorii do..." a zadáním názvu podkategorie; nebo pomocí Editoru uživatelských kategorií zadáním názvů, jako výše uvedený příklad Oblíbené.

V uživatelských kategoriích můžete vyhledávat stejným způsobem jako ve vestavěných kategoriích kliknutím na ně. Při kliknut

1. "všechno odpovídající položce v kategorii" označené jednoduchým zeleným znaménkem plus.
2. "všechno odpovídající položce v kategorii nebo jejím podkategoriím" označené dvěma zelenými znaménky plus.
3. "všechno neodpovídající položce v kategorii" označené jednoduchým červeným znaménkem minus.
4. "všechno neodpovídající položce v kategorii nebo jejím podkategoriím" označené dvěma červenými znaménky minus.

Je také možné vytvářet hierarchie uvnitř některých textových kategorií, jako jsou štítky, série a vlastní sloupce. Tyto hierarchie se zobrazují s malým trojúhelníkem umožňujícím skrytí podpoložek. Pro použití hierarchií položek v kategorii musíte nejprve přejít do Předvolby->Vzhled a zadat názvy kategorií do pole "Kategorie s hierarchickými položkami". Jakmile to uděláte, položky v této kategorii, které obsahují tečky, se zobrazí pomocí malého trojúhelníku. Například předpokládejme, že vytvoříte vlastní sloupec nazvaný "Žánr" a označíte ho, že obsahuje hierarchické položky. Jakmile to uděláte, položky jako Mysteriózní.Thriller a Mysteriózní.Česky se zobrazí jako Mysteriózní s malým trojúhelníkem. Kliknutí na trojúhelník zobrazí Thriller a Česky jako podpoložky. Pro více informací se podívejte na *Správa podskupiny knih, například "žánr"* (stránka 126).

Hierarchické položky (položek s podřízenými položkami) používají stejná čtyři vyhledávání 'kliknutím na' jako uživatelské kategorie. Položky, které nemají podřízené položky, dvě z vyhledávání: "všechno odpovídající" a "všechno neodpovídající".

Položky v Prohlížeči štítků můžete přetahovat na uživatelské kategorie pro jejich přidání do této kategorie. Pokud je zdrojem uživatelská kategorie, podržení klávesy Shift při přetažení přesune položku do nové kategorie. Můžete také přetahovat knihy ze seznamu knih na položky v Prohlížeči štítků; přetažení knihy na položku způsobí, že položka bude automaticky použita na přetaženou knihu. Například přetažení knihy na Isaaca Asimova nastaví autora této knihy na Isaaca Asimova. Její přetažení na štítek Historická přidá štítek Historická do štítků knihy.

V horní části Prohlížeče štítků je panel hledání, který vám umožňuje snadno najít jakoukoliv položku v Prohlížeči štítků. Navíc můžete kliknout pravým tlačítkem myši na kteroukoliv položku a zvolit jednu z několika operací. Některé příklady jsou skrýt ji, přejmenovat ji nebo otevřít dialogové okno "Spravovat x", které vám umožňuje spravovat položky tohoto druhu. Například dialogové okno "Spravovat autory" vám umožňuje přejmenovat autory a určit, jak jsou řazena jejich jména.

Jak jsou položky v Prohlížeči štítků řazeny, můžete určit pomocí tlačítka *Upravit Prohlížeč štítků* v dolní části Prohlížeče štítků. Můžete si zvolit seřazení podle názvu, průměrného hodnocení nebo oblíbenosti (oblíbenost je počet knih u položky ve vaší knihovně; například oblíbenost Isaaca Asimova je počet knih ve vaší knihovně od Isaaca Asimova).

1.1.11 Mřížka obálek

Můžete nechat Calibre zobrazit mřížku obálek knih namísto seznamu knih, pokud dáváte přednost procházení své kolekce podle obálek. *Mřížka obálek* se aktivuje kliknutím na tlačítko mřížky v pravém dolním rohu hlavního okna Calibre. Velikosti obálek a pozadí mřížky obálek můžete přizpůsobit pomocí *Předvolby->Vzhled->Mřížka obálek*. Můžete dokonce nechat Calibre zobrazit jakékoliv zadané pole pod obálkami podle své vlastní vůle, jako je název, autoři, hodnocení nebo vlastní sloupec.

1.1.12 Prohlížeč obálek

Kromě mřížky obálek popsané výše, můžete také nechat Calibre zobrazit obálky v jedné řadě. To se aktivuje pomocí tlačítka v pravém dolním rohu hlavního okna. V *Předvolby->Vzhled->Prohlížeč obálek* můžete změnit počet zobrazených obálek a dokonce nechat zobrazit samotný prohlížeč obálek v samostatném automaticky otevíraném okně.

1.1.13 Rychlé zobrazení

Někdy chcete vybrat knihu a rychle získat seznam knih se stejnou hodnotou v některé kategorii (autoři, štítky, vydavatel, série atd.) jako aktuálně vybraná kniha, ale beze změny aktuálního zobrazení knihovny. Můžete to udělat pomocí Rychlého zobrazení. Rychlé zobrazení otevře druhé okno zobrazující seznam knih odpovídajících hodnotě, o kterou máte zájem.

Předpokládejme například, že chcete vidět seznam všech knih se stejným autorem jako u aktuálně vybrané knihy. Klikněte do buňky autora, který vás zajímá, a stiskněte klávesu ‘Q’. Otevře se okno se všemi autory pro tuto knihu vlevo a všemi knihami od vybraného autora vpravo.

Některé příklady použití Rychlého zobrazení: rychlé ukázání, které další knihy:

- mají některý štítek, který je použit u aktuálně vybrané knihy,
- jsou ve stejné sérii jako aktuální kniha,
- mají stejné hodnoty ve vlastním sloupci jako aktuální kniha,
- jsou napsány jedním ze stejných autorů aktuální knihy

beze změny obsahu zobrazení knihovny.

Okno Rychlého zobrazení se otevře nad oknem Calibre a zůstane otevřené, dokud ho výslovně nezavřete. Rychlé zobrazení a zobrazení knihovny Calibre můžete použít současně. Například pokud v zobrazení knihovny Calibre kliknete na sloupec kategorie (štítky, série, vydavatel, autoři atd.) pro knihu, obsah okna Rychlého zobrazení se změní, aby vám ukázal v levém bočním podokně položky v kategorii pro vybranou knihu (např. štítky pro tuto knihu). Bude vybrána první položka v tomto seznamu a Rychlé zobrazení vám ukáže v pravém bočním podokně všechny knihy ve vaší knihovně, které odkazují na tuto položku. Klikněte na jinou položku v levém podokně, abyste viděli knihy s touto jinou položkou.

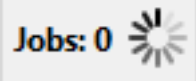
Poklepejte na knihu v okně Rychlého zobrazení pro výběr této knihy v zobrazení knihovny. Toto také změní zobrazení položek v okně Rychlého zobrazení (levé podokno), aby ukázalo položky v nově vybrané knize.

Poklepejte se Shift (nebo Ctrl) na knihu v okně Rychlého zobrazení pro otevření dialogového okna úpravy metadat této knihy v okně Calibre.

Přejetím myší nad záhlavím sloupce a podíváním se na popis tlačítka pro toto záhlaví můžete vidět, zda sloupec může být zobrazen v Rychlém zobrazení. Také můžete zjistit kliknutím pravým tlačítkem myši na záhlaví sloupce, zda je volba "Rychlé zobrazení" uvedeno v nabídce, v takovém případě výběr volby Rychlého zobrazení odpovídá stisknutí 'Q' v aktuální buňce.

Rychlé zobrazení respektuje nastavení virtuální knihovny, zobrazuje pouze knihy v aktuální virtuální knihovně.

1.1.14 Úlohy



Panel Úlohy zobrazuje počet aktuálně spuštěných úloh. Úlohy jsou úkoly, které běží v samostatném procesu. Zahrnují převody e-knih a komunikaci s vaším čtecím zařízení. Pro přístup k seznamu úloh můžete kliknout na panel úloh. Jakmile je úloha dokončena, můžete si prohlédnout podrobný protokol z této úlohy poklepáním na něj v seznamu. To je užitečné pro úlohy ladění, které nemusely být úspěšně dokončeny.

1.1.15 Klávesové zkratky

Calibre má několik klávesových zkratek, aby vám ušetřil čas a pohyb myší. Tyto zkratky jsou aktivní v zobrazení seznamu knih (pokud neupravujete podrobnosti konkrétní knihy) a většina z nich ovlivňuje název, který jste vybrali. Prohlížeč e-knih Calibre má své vlastní zkratky, které mohou být přizpůsobeny kliknutím na tlačítko Předvolby v prohlížeči.

Poznámka: Poznámka: Klávesové zkratky Calibre nevyžadují modifikační klávesu (Command, Option, Control atd.), pokud není výslovně uvedeno jinak. Stačí pouze stisknout klávesu s písmenem, například E pro úpravu.

Tabulka 1.1: [u'Kl1vesov9 zkratky']

Klávesová zkratka	Akce
`F2 (Enter v OS X)`	Upravit metadata aktuálně vybraného pole v seznamu knih
`A`	Přidat knihy
`Shift+A`	Přidat formáty k vybraným knihám
`C`	Převést vybrané knihy
`D`	Odeslat do zařízení
`Del`	Odebrat vybrané knihy
`E`	Upravit metadata vybraných knih
`G`	Získat knihy
`I`	Zobrazit podrobnosti o knize
`K`	Upravit obsah
`M`	Sloučit vybrané záznamy
`Alt+M`	Sloučit vybrané záznamy, ponechat originály
`O`	Otevřít nadřazenou složku
`P`	Vylepšit knihy
`S`	Uložit na disk

Tabulka 1.1 – pokračujte na předchozí stránce

Klávesová zkratka	Akce
T	Upravit knihu
V	Zobrazit
Alt+V nebo Cmd+V v OS X	Zobrazit konkrétní formát
Alt+Shift+J	Přepnout seznam úloh
Alt+Shift+B	Přepnout prohlížeč obálek
Alt+Shift+D	Přepnout panel podrobností o knize
Alt+Shift+T	Přepnout prohlížeč štítků
Alt+Shift+G	Přepnout mřížku obálek
Alt+A	Zobrazit knihy od stejného autora jako aktuální kniha
Alt+T	Zobrazit knihy se stejnými štítky jako aktuální kniha
Alt+P	Zobrazit knihy od stejného vydavatele jako aktuální kniha
Alt+Shift+S	Zobrazit knihy ze stejné série jako aktuální kniha
/, Ctrl+F	Zaměřit panel hledání
Shift+Ctrl+F	Otevřít dialogové okno pokročilého hledání
Esc	Vymazat aktuální hledání
Shift+Esc	Zaměřit seznam knih
Ctrl+Esc	Vymazat virtuální knihovnu
Alt+Esc	Vymazat dodatečná omezení
Ctrl+*	Vytvořit dočasnou virtuální knihovnu založenou na aktuálním hledání
Ctrl+Vpravo	Vybrat další kartu virtuální knihovny
Ctrl+Vlevo	Vybrat předchozí kartu virtuální knihovny
N nebo F3	Najít další knihu, která odpovídá aktuálnímu hledání (funguje, pouze pokud je zapnuto zvýrazňo
Shift+N nebo Shift+F3	Najít předchozí knihu, která odpovídá aktuálnímu hledání (funguje, pouze pokud je zapnuto zvý
Ctrl+D	Stáhnout metadata a obálky
Ctrl+R	Restartovat Calibre
Ctrl+Shift+R	Restartovat Calibre v režimu ladění
Shift+Ctrl+E	Přidat prázdné knihy do Calibre
Ctrl+M	Přepnout stav označení nebo neoznačení pro vybrané knihy
Q	Otevřít automaticky otevírané Rychlé zobrazení pro zobrazení knih v souvisejících sériích, štítcí
Shift+Q	Zaměřit na otevřený panel Rychlého zobrazení
Shift+S	Provést hledání v panelu Rychlého zobrazení
Ctrl+Q	Ukončit Calibre

1.2 Přidávání oblíbených webových stránek se zprávami

Calibre má výkonné, pružné a snadno použitelné rozhraní pro stahování zpráv z internetu a jejich převodu do e-knihy. Následující vám za pomoci příkladů ukáže, jak získat zprávy z různých webových stránek.

K pochopení, jak používat rozhraní, následujte příklady ve vypsaném pořadí níže:

- *Úplně automatické načítání* (stránka 24)
  - *Blog Calibre* (stránka 24)
  - *bbc.co.uk* (stránka 26)
- *Přizpůsobení procesu načítání* (stránka 26)
  - *Použití tiskové verze bbc.co.uk* (stránka 26)
  - *Nahrazování stylů článku* (stránka 27)
  - *Krájení a sekání* (stránka 28)
  - *Příklad ze skutečného života* (stránka 40)
- *Tipy pro vývoj nových předpisů* (stránka 42)
- *Další čtení* (stránka 43)
- *Dokumentace API* (stránka 43)

1.2.1 Úplně automatické načítání

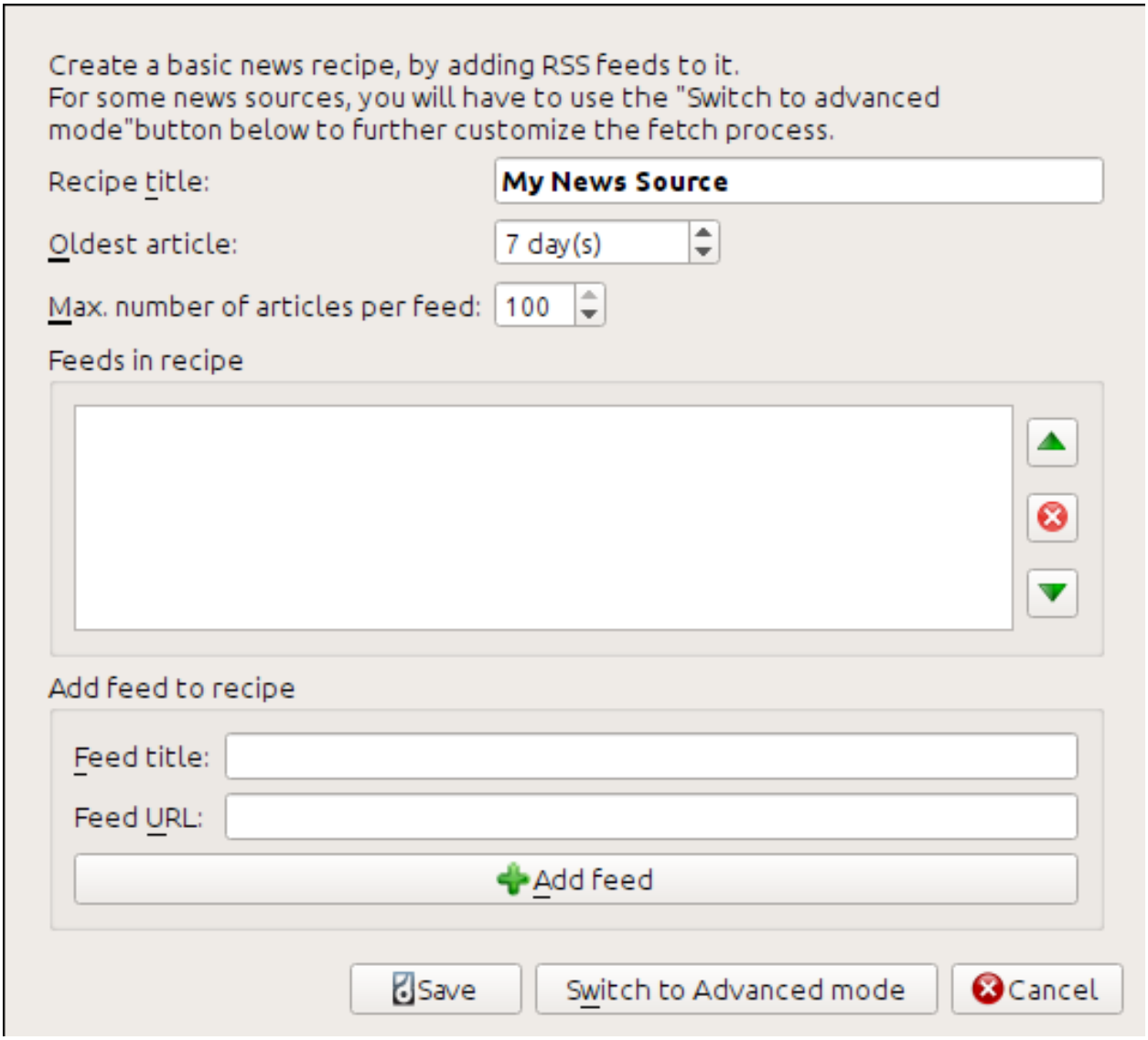
Pokud je váš zdroj zpráv dostatečně jednoduchý, Calibre může být schopné získat ho úplně automaticky, potřebujete jen zadat URL. Calibre shromáždí všechny informace potřebné ke stažení zdroje zpráv do předpisu. Abyste řekli Calibre o zdroji zpráv, musíte pro to vytvořit *předpis*. Pojďme se podívat na nějaké příklady:

Blog Calibre

Blog Calibre je blog příspěvků, které popisují mnoho užitečných funkcí Calibre jednoduchým a přístupným způsobem pro nové uživatele Calibre. Pro stažení tohoto blogu do e-knihy spoléháme na informační kanál *RSS* blogu:

```
http://blog.calibre-ebook.com/feeds/posts/default
```

URL RSS jsem získal podíváním se pod “Subscribe to” dole na stránce blogu a výběrem *Posts->Atom*. Aby Calibre stáhlo informační kanály a převedlo je na e-knihu, měli byste kliknout pravým tlačítkem myši na tlačítko *Načíst zprávy* a pak na položku nabídky *Přidat vlastní zdroj zpráv* a pak na tlačítko :guilabel:‘Nový předpis. Mělo by se otevřít dialogové okno podobné tomu níže zobrazenému.


Create a basic news recipe, by adding RSS feeds to it.
For some news sources, you will have to use the "Switch to advanced mode"button below to further customize the fetch process.
Recipe title:
My News Source
Oldest article:
7 day(s)
Max. number of articles per feed:
100
Feeds in recipe
Add feed to recipe
Feed title:
Feed URL:
Add feed
Save
Switch to Advanced mode
Cancel

Nejdříve zadejte `Blog Calibre` do pole *Název předpisu*. To bude název e-knihy, která bude vytvořena z článků ve výše uvedených informačních kanálech.

Další dvě pole (*Nejstarší článek* a *Max. počet článků*) vám umožní kontrolu nad tím, kolik článků by mělo být staženo z každého informačního kanálu, a jsou docela sebevysvětlující.

Pro přidání informačních kanálů do předpisu zadejte název a URL informačního kanálu a klikněte na tlačítko *Přidat informační kanál*. Jakmile jste přidali informační kanál, jednoduše klikněte na tlačítko *Uložit* a máte hotovo! Zavřete dialogové okno.

Pro otestování nového *předpisu* klikněte na tlačítko *Načíst zprávy* a v podnabídce *Vlastní zdroje zpráv* klikněte na *Blog Calibre*. Po několika minutách se nově stažená e-kniha příspěvků blogu zobrazí v hlavním zobrazení knihovny (pokud máte připojenou čtečku, bude umístěna do čtečky namísto do knihovny). Vyberte ji a stiskněte tlačítko *Zobrazit* pro čtení!

Důvod, proč toto fungovalo tak dobře s tak málo úsilím, je ten, že blog poskytuje informační kanály *RSS* s *úplným obsahem*, t.j. obsah článku je vložen do samotného informačního kanálu. Pro většinu zdrojů zpráv, které poskytují zprávy tímto způsobem s informačními kanály s *úplným obsahem*, nepotřebujete žádné další úsilí, abyste je převedli na e-knihy. Nyní se podíváme na zdroje zpráv, které neposkytují informační kanály s úplným obsahem. V takových informačních kanálech je celý článek webová stránka a informační kanál obsahuje pouze odkaz na webovou stránku s

krátkým souhrnem článku.

bbc.co.uk

Zkusme následující dva informační kanály z *BBC*:

1. Titulní sránka zpráv: http://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml
2. Věda a příroda: http://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

Postupujte podle pokynů navržených ve výše uvedeném *Blog Calibre* (stránka 24) pro vytvoření předpisu pro *BBC* (pomocí výše uvedených informačních kanálů). Při pohledu na staženou e-knihu vidíme, že Calibre udělalo slušnou práci vyextrahováním pouze obsahu, na kterém vám záleží, z každé webové stránky článku. Proces extrakce však není dokonalý. Někdy ponechá nežádoucí obsah, jako jsou nabídky a navigační pomůcky, nebo odebere obsah, který by měl být ponechán, jako jsou záhlaví článku. Abychom dosáhli dokonalé extrakce obsahu, budeme muset přizpůsobit proces načtení, jak je popsáno v následujícím oddíle.

1.2.2 Přizpůsobení procesu načítání

Pokud chcete zdokonalit proces stahování, nebo stáhnout obsah z obzvláště složité webové stránky, můžete využít celou sílu a pružnost rozhraní předpisů. Za tímto účelem v dialogovém okně *Přidat vlastní zdroj zpráv* jednoduše klikněte na tlačítko *Přepnout do pokročilého režimu*.

Nejjednodušší a často nejproduktivnější přizpůsobení je použít tiskovou verzi online článků. Tisková verze má obvykle mnohem méně nadbytečných dat a překládá se mnohem plynuleji do e-knihy. Zkusme použít tiskovou verzi článků z *BBC*.

Použití tiskové verze bbc.co.uk

Prvním krokem je podívat se na e-knihu, kterou jsme dříve stáhli z *bbc.co.uk* (stránka 26). Na konci každého článku v e-knize je malá záložka oznamující, odkud byl článek stažen z. Zkopírujte a vložte tuto URL do prohlížeče. Nyní na webové stránce článku najděte odkaz, který odkazuje na "Tisknutelnou verzi". Klikněte na něj pro zobrazení tiskové verze článku. Vypadá to mnohem čistěji! Nyní porovnejte obě URL. Pro mě byly:

URL článku http://news.bbc.co.uk/2/hi/science/nature/7312016.stm

URL tiskové verze http://newsvote.bbc.co.uk/mpapps/pagetools/print/news.bbc.co.uk/2/hi/science/nature/7312016.stm

Takže to vypadá, že pro získání tiskové verze musíme přidat ke každé URL článku předponu:

newsvote.bbc.co.uk/mpapps/pagetools/print/

Nyní byste měli v *Pokročilém režimu* dialogového okna Vlastní zdroje zpráv vidět něco podobného (nezapomeňte vybrat předpis *BBC* před přepnutím do pokročilého režimu):

Můžete vidět, že pole ze *Základního režimu* byla přeložena do kódu Pythonu přímočarým způsobem. Potřebujeme přidat pokyny k tomuto předpisu, aby použil tiskovou verzi článků. Vše, co je potřeba, je přidat následující dva řádky:

```
def print_version(self, url):
    return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/print/')
```

Toto je Python, takže odsazení je důležité. Poté, co jste přidali řádky, mělo by to vypadat takto:

Recipe source code (python)

```
class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonlir

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/p
```

Ve výše uvedeném `def print_version(self, url)` definuje *metodu*, která je volána Calibre pro každý článek. `url` je URL původního článku. `print_version` dělá to, že vezme tuto URL a nahradí ji novou URL, která odkazuje na tiskovou verzi článku. Abyste se naučili více o Pythonu[3], podívejte se na kurz[4].

Nyní klikněte na tlačítko *Přidat nebo aktualizovat předpis* a vaše změny budou uloženy. Znovu stáhněte e-knihu. Měli byste mít mnohem lepší e-knihu. Jeden z problémů s novou verzí je, že písma na tiskové verzi webové stránky jsou příliš malá. To je automaticky opraveno při převodu na e-knihu, ale i po procesu opravy bude velikost písma nabídek a navigačního panelu příliš velká vzhledem k textu článku. Abychom to opravili, provedeme nějaká další přizpůsobení v dalším oddíle.

Nahrazování stylů článku

V předchozím oddíle jsme viděli, že velikost písma pro články z tiskové verze *BBC* byla příliš malá. Ve většině webových stránek, včetně *BBC*, je tato velikost písma nastavena pomocí šablon stylů *CSS*. Můžeme zakázat načítání těchto šablon stylů přidáním řádku:

```
no_stylesheets = True
```

Předpis nyní vypadá takto:

[3] http://www.python.org
[4] https://docs.python.org/2/tutorial/

Recipe source code (python)

```
class AdvancedUserRecipe1206419520(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100
    no_stylesheets = True

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonli

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/
```

Nová verze vypadá docela dobře. Pokud jste perfekcionista, budete si chtít přečíst další oddíl, který se zabývá skutečnou úpravou staženého obsahu.

Krájení a sekání

Calibre obsahuje velice výkonné a pružné možnosti, pokud jde o zacházení se staženým obsahem. Abychom vám pár z nich ukázali, pojďme se znovu podívat na našeho starého přítele, na předpis *BBC* (stránka 27). Při pohledu na zdrojový kód (*HTML*) pár článků (tisková verze) vidíme, že mají patičku, která neobsahuje žádné užitečné informace, obsažené v

```
<div class="footer">
...
</div>
```

Toto může být odebráno přidáním:

```
remove_tags    = [dict(name='div', attrs={'class':'footer'})]
```

k předpisu. A nakonec nahraďme některé z *CSS*, které jsme dříve zakázali, naším vlastním *CSS*, které je vhodné pro převod na e-knihu:

```
extra_css      = '.headline {font-size: x-large;} \n .fact { padding-top: 10pt  }'
```

Pomocí těchto přídavků náš předpis získal "výrobní kvalitu", vlastně se velice blíží skutečnému předpisu používanému Calibre pro *BBC*, který je zobrazen níže:

```
##
## Title:         BBC News, Sport, and Blog Calibre Recipe
## Contact:       mattst - jmstanfield@gmail.com
##
## License:       GNU General Public License v3 - http://www.gnu.org/copyleft/gpl.html
## Copyright:     mattst - jmstanfield@gmail.com
##
## Written:       November 2011
## Last Edited:   2011-11-19
##

__license__     = 'GNU General Public License v3 - http://www.gnu.org/copyleft/gpl.html'
__copyright__   = 'mattst - jmstanfield@gmail.com'


'''
```

```
BBC News, Sport, and Blog Calibre Recipe
'''

# Import the regular expressions module.
import re

# Import the BasicNewsRecipe class which this class extends.
from calibre.web.feeds.recipes import BasicNewsRecipe

class BBCNewsSportBlog(BasicNewsRecipe):

    #
    #    **** IMPORTANT USERS READ ME ****
    #
    #  First select the feeds you want then scroll down below the feeds list
    #  and select the values you want for the other user preferences, like
    #  oldest_article and such like.
    #
    #
    #  Select the BBC rss feeds which you want in your ebook.
    #  Selected feed have NO '#' at their start, de-selected feeds begin with a '#'.
    #
    #  Eg.  ("News Home", "http://feeds.bbci.co.uk/... - include feed.
    #  Eg. #("News Home", "http://feeds.bbci.co.uk/... - do not include feed.
    #
    # There are 68 feeds below which constitute the bulk of the available rss
    # feeds on the BBC web site. These include 5 blogs by editors and
    # correspondants, 16 sports feeds, 15 'sub' regional feeds (Eg. North West
    # Wales, Scotland Business), and 7 Welsh language feeds.
    #
    # Some of the feeds are low volume (Eg. blogs), or very low volume (Eg. Click)
    # so if "oldest_article = 1.5" (only articles published in the last 36 hours)
    # you may get some 'empty feeds' which will not then be included in the ebook.
    #
    # The 15 feeds currently selected below are simply my default ones.
    #
    # Note: With all 68 feeds selected, oldest_article set to 2,
    # max_articles_per_feed set to 100, and simultaneous_downloads set to 10,
    # the ebook creation took 29 minutes on my speedy 100 mbps net connection,
    # fairly high-end desktop PC running Linux (Ubuntu Lucid-Lynx).
    # More realistically with 15 feeds selected, oldest_article set to 1.5,
    # max_articles_per_feed set to 100, and simultaneous_downloads set to 20,
    # it took 6 minutes. If that's too slow increase 'simultaneous_downloads'.
    #
    # Select / de-select the feeds you want in your ebook.
    #
    feeds = [
              ("News Home", "http://feeds.bbci.co.uk/news/rss.xml"),
              ("UK", "http://feeds.bbci.co.uk/news/uk/rss.xml"),
              ("World", "http://feeds.bbci.co.uk/news/world/rss.xml"),
              #("England", "http://feeds.bbci.co.uk/news/england/rss.xml"),
              #("Scotland", "http://feeds.bbci.co.uk/news/scotland/rss.xml"),
              #("Wales", "http://feeds.bbci.co.uk/news/wales/rss.xml"),
              #("N. Ireland", "http://feeds.bbci.co.uk/news/northern_ireland/rss.xml"),
              #("Africa", "http://feeds.bbci.co.uk/news/world/africa/rss.xml"),
              #("Asia", "http://feeds.bbci.co.uk/news/world/asia/rss.xml"),
              #("Europe", "http://feeds.bbci.co.uk/news/world/europe/rss.xml"),
              #("Latin America", "http://feeds.bbci.co.uk/news/world/latin_america/rss.xml"),
```

```
            #("Middle East", "http://feeds.bbci.co.uk/news/world/middle_east/rss.xml"),
            ("US & Canada", "http://feeds.bbci.co.uk/news/world/us_and_canada/rss.xml"),
            ("Politics", "http://feeds.bbci.co.uk/news/politics/rss.xml"),
            ("Science/Environment", "http://feeds.bbci.co.uk/news/science_and_environment/rss.xml"),
            ("Technology", "http://feeds.bbci.co.uk/news/technology/rss.xml"),
            ("Magazine", "http://feeds.bbci.co.uk/news/magazine/rss.xml"),
            ("Entertainment/Arts", "http://feeds.bbci.co.uk/news/entertainment_and_arts/rss.xml"),
            #("Health", "http://feeds.bbci.co.uk/news/health/rss.xml"),
            #("Education/Family", "http://feeds.bbci.co.uk/news/education/rss.xml"),
            ("Business", "http://feeds.bbci.co.uk/news/business/rss.xml"),
            ("Special Reports", "http://feeds.bbci.co.uk/news/special_reports/rss.xml"),
            ("Also in the News", "http://feeds.bbci.co.uk/news/also_in_the_news/rss.xml"),
            #("Newsbeat", "http://www.bbc.co.uk/newsbeat/rss.xml"),
            #("Click", "http://newsrss.bbc.co.uk/rss/newsonline_uk_edition/programmes/click_online/
            ("Blog: Nick Robinson (Political Editor)", "http://feeds.bbci.co.uk/news/correspondents
            #("Blog: Mark D'Arcy (Parliamentary Correspondent)", "http://feeds.bbci.co.uk/news/cor
            #("Blog: Robert Peston (Business Editor)", "http://feeds.bbci.co.uk/news/correspondents
            #("Blog: Stephanie Flanders (Economics Editor)", "http://feeds.bbci.co.uk/news/corresp
            ("Blog: Rory Cellan-Jones (Technology correspondent)", "http://feeds.bbci.co.uk/news/co
            ("Sport Front Page", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/front_page/rs
            #("Football", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/football/rss.xml"),
            #("Cricket", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/cricket/rss.xml"),
            #("Rugby Union", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/rugby_union/rss.x
            #("Rugby League", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/rugby_league/rss
            #("Tennis", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/tennis/rss.xml"),
            #("Golf", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/golf/rss.xml"),
            #("Motorsport", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/motorsport/rss.xml
            #("Boxing", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/boxing/rss.xml"),
            #("Athletics", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/athletics/rss.xml",
            #("Snooker", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports/snooker/
            #("Horse Racing", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports/hor
            #("Cycling", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports/cycling/
            #("Disability Sport", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports
            #("Other Sport", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports/rss
            #("Olympics 2012", "http://newsrss.bbc.co.uk/rss/sportonline_uk_edition/other_sports/ol
            #("N. Ireland Politics", "http://feeds.bbci.co.uk/news/northern_ireland/northern_irelan
            #("Scotland Politics", "http://feeds.bbci.co.uk/news/scotland/scotland_politics/rss.xml
            #("Scotland Business", "http://feeds.bbci.co.uk/news/scotland/scotland_business/rss.xml
            #("E. Scotland, Edinburgh & Fife", "http://feeds.bbci.co.uk/news/scotland/edinburgh_eas
            #("W. Scotland & Glasgow", "http://feeds.bbci.co.uk/news/scotland/glasgow_and_west/rss
            #("Highlands & Islands", "http://feeds.bbci.co.uk/news/scotland/highlands_and_islands/r
            #("NE. Scotland, Orkney & Shetland", "http://feeds.bbci.co.uk/news/scotland/north_east_
            #("South Scotland", "http://feeds.bbci.co.uk/news/scotland/south_scotland/rss.xml"),
            #("Central Scotland & Tayside", "http://feeds.bbci.co.uk/news/scotland/tayside_and_cent
            #("Wales Politics", "http://feeds.bbci.co.uk/news/wales/wales_politics/rss.xml"),
            #("NW. Wales", "http://feeds.bbci.co.uk/news/wales/north_west_wales/rss.xml"),
            #("NE. Wales", "http://feeds.bbci.co.uk/news/wales/north_east_wales/rss.xml"),
            #("Mid. Wales", "http://feeds.bbci.co.uk/news/wales/mid_wales/rss.xml"),
            #("SW. Wales", "http://feeds.bbci.co.uk/news/wales/south_west_wales/rss.xml"),
            #("SE. Wales", "http://feeds.bbci.co.uk/news/wales/south_east_wales/rss.xml"),
            #("Newyddion - News in Welsh", "http://feeds.bbci.co.uk/newyddion/rss.xml"),
            #("Gwleidyddiaeth", "http://feeds.bbci.co.uk/newyddion/gwleidyddiaeth/rss.xml"),
            #("Gogledd-Ddwyrain", "http://feeds.bbci.co.uk/newyddion/gogledd-ddwyrain/rss.xml"),
            #("Gogledd-Orllewin", "http://feeds.bbci.co.uk/newyddion/gogledd-orllewin/rss.xml"),
            #("Canolbarth", "http://feeds.bbci.co.uk/newyddion/canolbarth/rss.xml"),
            #("De-Ddwyrain", "http://feeds.bbci.co.uk/newyddion/de-ddwyrain/rss.xml"),
            #("De-Orllewin", "http://feeds.bbci.co.uk/newyddion/de-orllewin/rss.xml"),
        ]
```

```
#    **** SELECT YOUR USER PREFERENCES ****

# Title to use for the ebook.
#
title = 'BBC News'

# A brief description for the ebook.
#
description = u'BBC web site ebook created using rss feeds.'

# The max number of articles which may be downloaded from each feed.
# I've never seen more than about 70 articles in a single feed in the
# BBC feeds.
#
max_articles_per_feed = 100

# The max age of articles which may be downloaded from each feed. This is
# specified in days - note fractions of days are allowed, Eg. 2.5 (2 and a
# half days). My default of 1.5 days is the last 36 hours, the point at
# which I've decided 'news' becomes 'old news', but be warned this is not
# so good for the blogs, technology, magazine, etc., and sports feeds.
# You may wish to extend this to 2-5 but watch out ebook creation time will
# increase as well. Setting this to 30 will get everything (AFAICT) as long
# as max_articles_per_feed remains set high (except for 'Click' which is
# v. low volume and its currently oldest article is 4th Feb 2011).
#
oldest_article = 1.5

# Number of simultaneous downloads. 20 is consistantly working fine on the
# BBC News feeds with no problems. Speeds things up from the defualt of 5.
# If you have a lot of feeds and/or have increased oldest_article above 2
# then you may wish to try increasing simultaneous_downloads to 25-30,
# Or, of course, if you are in a hurry. [I've not tried beyond 20.]
#
simultaneous_downloads = 20

# Timeout for fetching files from the server in seconds. The default of
# 120 seconds, seems somewhat excessive.
#
timeout = 30

# The format string for the date shown on the ebook's first page.
# List of all values: http://docs.python.org/library/time.html
# Default in news.py has a leading space so that's mirrored here.
# As with 'feeds' select/de-select by adding/removing the initial '#',
# only one timefmt should be selected, here's a few to choose from.
#
timefmt = ' [%a, %d %b %Y]'              # [Fri, 14 Nov 2011] (Calibre default)
#timefmt = ' [%a, %d %b %Y %H:%M]'       # [Fri, 14 Nov 2011 18:30]
#timefmt = ' [%a, %d %b %Y %I:%M %p]'    # [Fri, 14 Nov 2011 06:30 PM]
#timefmt = ' [%d %b %Y]'                 # [14 Nov 2011]
#timefmt = ' [%d %b %Y %H:%M]'           # [14 Nov 2011 18.30]
#timefmt = ' [%Y-%m-%d]'                 # [2011-11-14]
#timefmt = ' [%Y-%m-%d-%H-%M]'           # [2011-11-14-18-30]
```

```
    #
    #     **** IMPORTANT ****
    #
    #     DO NOT EDIT BELOW HERE UNLESS YOU KNOW WHAT YOU ARE DOING.
    #
    #     DO NOT EDIT BELOW HERE UNLESS YOU KNOW WHAT YOU ARE DOING.
    #
    #     I MEAN IT, YES I DO, ABSOLUTELY, AT YOU OWN RISK. :)
    #
    #     **** IMPORTANT ****
    #



    # Author of this recipe.
    __author__ = 'mattst'

    # Specify English as the language of the RSS feeds (ISO-639 code).
    language = 'en_GB'

    # Set tags.
    tags = 'news, sport, blog'

    # Set publisher and publication type.
    publisher = 'BBC'
    publication_type = 'newspaper'

    # Disable stylesheets from site.
    no_stylesheets = True

    # Specifies an override encoding for sites that have an incorrect charset
    # specified. Default of 'None' says to auto-detect. Some other BBC recipes
    # use 'utf8', which works fine (so use that if necessary) but auto-detecting
    # with None is working fine, so stick with that for robustness.
    encoding = None

    # Sets whether a feed has full articles embedded in it. The BBC feeds do not.
    use_embedded_content = False

    # Removes empty feeds - why keep them!?
    remove_empty_feeds = True

    # Create a custom title which fits nicely in the Kindle title list.
    # Requires "import time" above class declaration, and replacing
    # title with custom_title in conversion_options (right column only).
    # Example of string below: "BBC News - 14 Nov 2011"
    #
    # custom_title = "BBC News - " + time.strftime('%d %b %Y')

    '''
    # Conversion options for advanced users, but don't forget to comment out the
    # current conversion_options below. Avoid setting 'linearize_tables' as that
    # plays havoc with the 'old style' table based pages.
    #
    conversion_options = { 'title'       : title,
                           'comments'    : description,
                           'tags'        : tags,
                           'language'    : language,
```

```
                        'publisher'   : publisher,
                        'authors'     : publisher,
                        'smarten_punctuation' : True
                      }
    '''

    conversion_options = { 'smarten_punctuation' : True }

    # Specify extra CSS - overrides ALL other CSS (IE. Added last).
    extra_css = 'body { font-family: verdana, helvetica, sans-serif; } \
                 .introduction, .first { font-weight: bold; } \
                 .cross-head { font-weight: bold; font-size: 125%; } \
                 .cap, .caption { display: block; font-size: 80%; font-style: italic; } \
                 .cap, .caption, .caption img, .caption span { display: block; text-align: center; ma
                 .byl, .byd, .byline img, .byline-name, .byline-title, .author-name, .author-positio
                    .correspondent-portrait img, .byline-lead-in, .name, .bbc-role { display: block;
                    text-align: center; font-size: 80%; font-style: italic; margin: 1px auto; } \
                 .story-date, .published { font-size: 80%; } \
                 table { width: 100%; } \
                 td img { display: block; margin: 5px auto; } \
                 ul { padding-top: 10px; } \
                 ol { padding-top: 10px; } \
                 li { padding-top: 5px; padding-bottom: 5px; } \
                 h1 { text-align: center; font-size: 175%; font-weight: bold; } \
                 h2 { text-align: center; font-size: 150%; font-weight: bold; } \
                 h3 { text-align: center; font-size: 125%; font-weight: bold; } \
                 h4, h5, h6 { text-align: center; font-size: 100%; font-weight: bold; }'

    # Remove various tag attributes to improve the look of the ebook pages.
    remove_attributes = [ 'border', 'cellspacing', 'align', 'cellpadding', 'colspan',
                          'valign', 'vspace', 'hspace', 'alt', 'width', 'height' ]

    # Remove the (admittedly rarely used) line breaks, "<br />", which sometimes
    # cause a section of the ebook to start in an unsightly fashion or, more
    # frequently, a "<br />" will muck up the formatting of a correspondant's byline.
    # "<br />" and "<br clear/>" are far more frequently used on the table formatted
    # style of pages, and really spoil the look of the ebook pages.
    preprocess_regexps     = [(re.compile(r'<br[ ]*/>', re.IGNORECASE), lambda m: ''),
                              (re.compile(r'<br[ ]*clear.*/>', re.IGNORECASE), lambda m: '')]


    # Create regular expressions for tag keeping and removal to make the matches more
    # robust against minor changes and errors in the HTML, Eg. double spaces, leading
    # and trailing spaces, missing hyphens, and such like.
    # Python regular expression ('re' class) page: http://docs.python.org/library/re.html

    # ***************************************
    # Regular expressions for keep_only_tags:
    # ***************************************

    # The BBC News HTML pages use variants of 'storybody' to denote the section of a HTML
    # page which contains the main text of the article. Match storybody variants: 'storybody',
    # 'story-body', 'story body','storybody ', etc.
    storybody_reg_exp = '^.*story[_ -]*body.*$'

    # The BBC sport and 'newsbeat' (features) HTML pages use 'blq_content' to hold the title
    # and published date. This is one level above the usual news pages which have the title
    # and date within 'story-body'. This is annoying since 'blq_content' must also be kept,
```

```
    # resulting in a lot of extra things to be removed by remove_tags.
    blq_content_reg_exp = '^.*blq[_ -]*content.*$'

    # The BBC has an alternative page design structure, which I suspect is an out-of-date
    # design but which is still used in some articles, Eg. 'Click' (technology), 'FastTrack'
    # (travel), and in some sport pages. These alternative pages are table based (which is
    # why I think they are an out-of-date design) and account for -I'm guesstimaking- less
    # than 1% of all articles. They use a table class 'storycontent' to hold the article
    # and like blq_content (above) have required lots of extra removal by remove_tags.
    story_content_reg_exp = '^.*story[_ -]*content.*$'

    # Keep the sections of the HTML which match the list below. The HTML page created by
    # Calibre will fill <body> with those sections which are matched. Note that the
    # blq_content_reg_exp must be listed before storybody_reg_exp in keep_only_tags due to
    # it being the parent of storybody_reg_exp, that is to say the div class/id 'story-body'
    # will be inside div class/id 'blq_content' in the HTML (if 'blq_content' is there at
    # all). If they are the other way around in keep_only_tags then blq_content_reg_exp
    # will end up being discarded.
    keep_only_tags = [ dict(name='table', attrs={'class':re.compile(story_content_reg_exp, re.IGNOREC
                       dict(name='div',   attrs={'class':re.compile(blq_content_reg_exp, re.IGNORECAS
                       dict(name='div',   attrs={'id':re.compile(blq_content_reg_exp, re.IGNORECASE)
                       dict(name='div',   attrs={'class':re.compile(storybody_reg_exp, re.IGNORECASE)
                       dict(name='div',   attrs={'id':re.compile(storybody_reg_exp, re.IGNORECASE)})

    # ***********************************
    # Regular expressions for remove_tags:
    # ***********************************

    # Regular expression to remove share-help and variant tags. The share-help class
    # is used by the site for a variety of 'sharing' type links, Eg. Facebook, delicious,
    # twitter, email. Removed to avoid page clutter.
    share_help_reg_exp = '^.*share[_ -]*help.*$'

    # Regular expression to remove embedded-hyper and variant tags. This class is used to
    # display links to other BBC News articles on the same/similar subject.
    embedded_hyper_reg_exp = '^.*embed*ed[_ -]*hyper.*$'

    # Regular expression to remove hypertabs and variant tags. This class is used to
    # display a tab bar at the top of an article which allows the user to switch to
    # an article (viewed on the same page) providing further info., 'in depth' analysis,
    # an editorial, a correspondant's blog entry, and such like. The ability to handle
    # a tab bar of this nature is currently beyond the scope of this recipe and
    # possibly of Calibre itself (not sure about that - TO DO - check!).
    hypertabs_reg_exp = '^.*hyper[_ -]*tabs.*$'

    # Regular expression to remove story-feature and variant tags. Eg. 'story-feature',
    # 'story-feature related narrow', 'story-feature wide', 'story-feature narrow'.
    # This class is used to add additional info. boxes, or small lists, outside of
    # the main story. TO DO: Work out a way to incorporate these neatly.
    story_feature_reg_exp = '^.*story[_ -]*feature.*$'

    # Regular expression to remove video and variant tags, Eg. 'videoInStoryB',
    # 'videoInStoryC'. This class is used to embed video.
    video_reg_exp = '^.*video.*$'

    # Regular expression to remove audio and variant tags, Eg. 'audioInStoryD'.
    # This class is used to embed audio.
    audio_reg_exp = '^.*audio.*$'
```

```
    # Regular expression to remove pictureGallery and variant tags, Eg. 'pictureGallery'.
    # This class is used to embed a photo slideshow. See also 'slideshow' below.
    picture_gallery_reg_exp = '^.*picture.*$'

    # Regular expression to remove slideshow and variant tags, Eg. 'dslideshow-enclosure'.
    # This class is used to embed a slideshow (not necessarily photo) but both
    # 'slideshow' and 'pictureGallery' are used for slideshows.
    slideshow_reg_exp = '^.*slide[_ -]*show.*$'

    # Regular expression to remove social-links and variant tags. This class is used to
    # display links to a BBC bloggers main page, used in various columnist's blogs
    # (Eg. Nick Robinson, Robert Preston).
    social_links_reg_exp = '^.*social[_ -]*links.*$'

    # Regular expression to remove quote and (multi) variant tags, Eg. 'quote',
    # 'endquote', 'quote-credit', 'quote-credit-title', etc. These are usually
    # removed by 'story-feature' removal (as they are usually within them), but
    # not always. The quotation removed is always (AFAICT) in the article text
    # as well but a 2nd copy is placed in a quote tag to draw attention to it.
    # The quote class tags may or may not appear in div's.
    quote_reg_exp = '^.*quote.*$'

    # Regular expression to remove hidden and variant tags, Eg. 'hidden'.
    # The purpose of these is unclear, they seem to be an internal link to a
    # section within the article, but the text of the link (Eg. 'Continue reading
    # the main story') never seems to be displayed anyway. Removed to avoid clutter.
    # The hidden class tags may or may not appear in div's.
    hidden_reg_exp = '^.*hidden.*$'

    # Regular expression to remove comment and variant tags, Eg. 'comment-introduction'.
    # Used on the site to display text about registered users entering comments.
    comment_reg_exp = '^.*comment.*$'

    # Regular expression to remove form and variant tags, Eg. 'comment-form'.
    # Used on the site to allow registered BBC users to fill in forms, typically
    # for entering comments about an article.
    form_reg_exp = '^.*form.*$'

    # Extra things to remove due to the addition of 'blq_content' in keep_only_tags.

    #<div class="story-actions"> Used on sports pages for 'email' and 'print'.
    story_actions_reg_exp = '^.*story[_ -]*actions.*$'

    #<div class="bookmark-list"> Used on sports pages instead of 'share-help' (for
    # social networking links).
    bookmark_list_reg_exp = '^.*bookmark[_ -]*list.*$'

    #<div id="secondary-content" class="content-group">
    # NOTE: Don't remove class="content-group" that is needed.
    # Used on sports pages to link to 'similar stories'.
    secondary_content_reg_exp = '^.*secondary[_ -]*content.*$'

    #<div id="featured-content" class="content-group">
    # NOTE: Don't remove class="content-group" that is needed.
    # Used on sports pages to link to pages like 'tables', 'fixtures', etc.
    featured_content_reg_exp = '^.*featured[_ -]*content.*$'

    #<div id="navigation">
```

```
    # Used on sports pages to link to pages like 'tables', 'fixtures', etc.
    # Used sometimes instead of "featured-content" above.
    navigation_reg_exp = '^.*navigation.*$'

    #<a class="skip" href="#blq-container-inner">Skip to top</a>
    # Used on sports pages to link to the top of the page.
    skip_reg_exp = '^.*skip.*$'

    # Extra things to remove due to the addition of 'storycontent' in keep_only_tags,
    # which are the alterative table design based pages. The purpose of some of these
    # is not entirely clear from the pages (which are a total mess!).

    # Remove mapping based tags, Eg. <map id="world_map">
    # The dynamic maps don't seem to work during ebook creation. TO DO: Investigate.
    map_reg_exp = '^.*map.*$'

    # Remove social bookmarking variation, called 'socialBookMarks'.
    social_bookmarks_reg_exp = '^.*social[_ -]*bookmarks.*$'

    # Remove page navigation tools, like 'search', 'email', 'print', called 'blq-mast'.
    blq_mast_reg_exp = '^.*blq[_ -]*mast.*$'

    # Remove 'sharesb', I think this is a generic 'sharing' class. It seems to appear
    # alongside 'socialBookMarks' whenever that appears. I am removing it as well
    # under the assumption that it can appear alone as well.
    sharesb_reg_exp = '^.*sharesb.*$'

    # Remove class 'o'. The worst named user created css class of all time. The creator
    # should immediately be fired. I've seen it used to hold nothing at all but with
    # 20 or so empty lines in it. Also to hold a single link to another article.
    # Whatever it was designed to do it is not wanted by this recipe. Exact match only.
    o_reg_exp = '^o$'

    # Remove 'promotopbg' and 'promobottombg', link lists. Have decided to
    # use two reg expressions to make removing this (and variants) robust.
    promo_top_reg_exp = '^.*promotopbg.*$'
    promo_bottom_reg_exp = '^.*promobottombg.*$'

    # Remove 'nlp', provides heading for link lists. Requires an exact match due to
    # risk of matching those letters in something needed, unless I see a variation
    # of 'nlp' used at a later date.
    nlp_reg_exp = '^nlp$'

    # Remove 'mva', provides embedded floating content of various types. Variant 'mvb'
    # has also now been seen. Requires an exact match of 'mva' or 'mvb' due to risk of
    # matching those letters in something needed.
    mva_or_mvb_reg_exp = '^mv[ab]$'

    # Remove 'mvtb', seems to be page navigation tools, like 'blq-mast'.
    mvtb_reg_exp = '^mvtb$'

    # Remove 'blq-toplink', class to provide a link to the top of the page.
    blq_toplink_reg_exp = '^.*blq[_ -]*top[_ -]*link.*$'

    # Remove 'products and services' links, Eg. desktop tools, alerts, and so on.
    # Eg. Class="servicev4 ukfs_services" - what a mess of a name. Have decided to
    # use two reg expressions to make removing this (and variants) robust.
    prods_services_01_reg_exp = '^.*servicev4.*$'
```

```
    prods_services_02_reg_exp = '^.*ukfs[_ -]*services.*$'

    # Remove -what I think is- some kind of navigation tools helper class, though I am
    # not sure, it's called: 'blq-rst blq-new-nav'. What I do know is it pops up
    # frequently and it is not wanted. Have decided to use two reg expressions to make
    # removing this (and variants) robust.
    blq_misc_01_reg_exp = '^.*blq[_ -]*rst.*$'
    blq_misc_02_reg_exp = '^.*blq[_ -]*new[_ -]*nav.*$'

    # Remove 'puffbox' - this may only appear inside 'storyextra', so it may not
    # need removing - I have no clue what it does other than it contains links.
    # Whatever it is - it is not part of the article and is not wanted.
    puffbox_reg_exp = '^.*puffbox.*$'

    # Remove 'sibtbg' and 'sibtbgf' - some kind of table formatting classes.
    sibtbg_reg_exp = '^.*sibtbg.*$'

    # Remove 'storyextra' - links to relevant articles and external sites.
    storyextra_reg_exp = '^.*story[_ -]*extra.*$'


    remove_tags = [ dict(name='div',  attrs={'class':re.compile(story_feature_reg_exp, re.IGNORECASE)
                    dict(name='div',  attrs={'class':re.compile(share_help_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(embedded_hyper_reg_exp, re.IGNORECASE
                    dict(name='div',  attrs={'class':re.compile(hypertabs_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(video_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(audio_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(picture_gallery_reg_exp, re.IGNORECAS
                    dict(name='div',  attrs={'class':re.compile(slideshow_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(quote_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(hidden_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(comment_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(story_actions_reg_exp, re.IGNORECASE)
                    dict(name='div',  attrs={'class':re.compile(bookmark_list_reg_exp, re.IGNORECASE)
                    dict(name='div',  attrs={'id':re.compile(secondary_content_reg_exp, re.IGNORECASE
                    dict(name='div',  attrs={'id':re.compile(featured_content_reg_exp, re.IGNORECASE)
                    dict(name='div',  attrs={'id':re.compile(navigation_reg_exp, re.IGNORECASE)}),
                    dict(name='form', attrs={'id':re.compile(form_reg_exp, re.IGNORECASE)}),
                    dict(attrs={'class':re.compile(quote_reg_exp, re.IGNORECASE)}),
                    dict(attrs={'class':re.compile(hidden_reg_exp, re.IGNORECASE)}),
                    dict(attrs={'class':re.compile(social_links_reg_exp, re.IGNORECASE)}),
                    dict(attrs={'class':re.compile(comment_reg_exp, re.IGNORECASE)}),
                    dict(attrs={'class':re.compile(skip_reg_exp, re.IGNORECASE)}),
                    dict(name='map', attrs={'id':re.compile(map_reg_exp, re.IGNORECASE)}),
                    dict(name='map', attrs={'name':re.compile(map_reg_exp, re.IGNORECASE)}),
                    dict(name='div', attrs={'id':re.compile(social_bookmarks_reg_exp, re.IGNORECASE)
                    dict(name='div', attrs={'id':re.compile(blq_mast_reg_exp, re.IGNORECASE)}),
                    dict(name='div', attrs={'class':re.compile(sharesb_reg_exp, re.IGNORECASE)}),
                    dict(name='div', attrs={'class':re.compile(o_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(promo_top_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(promo_bottom_reg_exp, re.IGNORECASE)
                    dict(name='div',  attrs={'class':re.compile(nlp_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(mva_or_mvb_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(mvtb_reg_exp, re.IGNORECASE)}),
                    dict(name='div',  attrs={'class':re.compile(blq_toplink_reg_exp, re.IGNORECASE)}
                    dict(name='div',  attrs={'class':re.compile(prods_services_01_reg_exp, re.IGNOREC
                    dict(name='div',  attrs={'class':re.compile(prods_services_02_reg_exp, re.IGNOREC
                    dict(name='div',  attrs={'class':re.compile(blq_misc_01_reg_exp, re.IGNORECASE)}
```

```
                dict(name='div',  attrs={'class':re.compile(blq_misc_02_reg_exp, re.IGNORECASE)}),
                dict(name='div',  attrs={'class':re.compile(puffbox_reg_exp, re.IGNORECASE)}),
                dict(attrs={'class':re.compile(sibtbg_reg_exp, re.IGNORECASE)}),
                dict(attrs={'class':re.compile(storyextra_reg_exp, re.IGNORECASE)})
            ]

    # Uses url to create and return the 'printer friendly' version of the url.
    # In other words the 'print this page' address of the page.
    #
    # There are 3 types of urls used in the BBC site's rss feeds. There is just
    # 1 type for the standard news while there are 2 used for sports feed urls.
    # Note: Sports urls are linked from regular news feeds (Eg. 'News Home') when
    # there is a major story of interest to 'everyone'. So even if no BBC sports
    # feeds are added to 'feeds' the logic of this method is still needed to avoid
    # blank / missing / empty articles which have an index title and then no body.
    def print_version(self, url):

        # Handle sports page urls type 01:
        if (url.find("go/rss/-/sport1/") != -1):
            temp_url = url.replace("go/rss/-/", "")

        # Handle sports page urls type 02:
        elif (url.find("go/rss/int/news/-/sport1/") != -1):
            temp_url = url.replace("go/rss/int/news/-/", "")

        # Handle regular news page urls:
        else:
            temp_url = url.replace("go/rss/int/news/-/", "")

        # Always add "?print=true" to the end of the url.
        print_url = temp_url + "?print=true"

        return print_url


    # Remove articles in feeds based on a string in the article title or url.
    #
    # Code logic written by: Starson17 - posted in: "Recipes - Re-usable code"
    # thread, in post with title: "Remove articles from feed", see url:
    # http://www.mobileread.com/forums/showpost.php?p=1165462&postcount=6
    # Many thanks and all credit to Starson17.
    #
    # Starson17's code has obviously been altered to suite my requirements.
    def parse_feeds(self):

        # Call parent's method.
        feeds = BasicNewsRecipe.parse_feeds(self)

        # Loop through all feeds.
        for feed in feeds:

            # Loop through all articles in feed.
            for article in feed.articles[:]:

                # Match key words and remove article if there's a match.

                # Most BBC rss feed video only 'articles' use upper case 'VIDEO'
                # as a title prefix. Just match upper case 'VIDEO', so that
```

```
                # articles like 'Video game banned' won't be matched and removed.
                if 'VIDEO' in article.title:
                    feed.articles.remove(article)

                # Most BBC rss feed audio only 'articles' use upper case 'AUDIO'
                # as a title prefix. Just match upper case 'AUDIO', so that
                # articles like 'Hi-Def audio...' won't be matched and removed.
                elif 'AUDIO' in article.title:
                    feed.articles.remove(article)

                # Most BBC rss feed photo slideshow 'articles' use 'In Pictures',
                # 'In pictures', and 'in pictures', somewhere in their title.
                # Match any case of that phrase.
                elif 'IN PICTURES' in article.title.upper():
                    feed.articles.remove(article)

                # As above, but user contributed pictures. Match any case.
                elif 'YOUR PICTURES' in article.title.upper():
                    feed.articles.remove(article)

                # 'Sportsday Live' are articles which contain a constantly and
                # dynamically updated 'running commentary' during a live sporting
                # event. Match any case.
                elif 'SPORTSDAY LIVE' in article.title.upper():
                    feed.articles.remove(article)

                # Sometimes 'Sportsday Live' (above) becomes 'Live - Sport Name'.
                # These are being matched below using 'Live - ' because removing all
                # articles with 'live' in their titles would remove some articles
                # that are in fact not live sports pages. Match any case.
                elif 'LIVE - ' in article.title.upper():
                    feed.articles.remove(article)

                # 'Quiz of the week' is a Flash player weekly news quiz. Match only
                # the 'Quiz of the' part in anticipation of monthly and yearly
                # variants. Match any case.
                elif 'QUIZ OF THE' in article.title.upper():
                    feed.articles.remove(article)

                # Remove articles with 'scorecards' in the url. These are BBC sports
                # pages which just display a cricket scorecard. The pages have a mass
                # of table and css entries to display the scorecards nicely. Probably
                # could make them work with this recipe, but might take a whole day
                # of work to sort out all the css - basically a formatting nightmare.
                elif 'scorecards' in article.url:
                    feed.articles.remove(article)

        return feeds

# End of class and file.
```

Tento *předpis* zkoumá pouze špičku ledovce, pokud jde o sílu Calibre. Abychom probádali více možností Calibre, prozkoumáme v dalším oddíle složitější příklad ze skutečného života.

Příklad ze skutečného života

Přiměřeně složitý příklad ze skutečného života, který odhaluje více z *API* `BasicNewsRecipe`, je *předpis* pro *The New York Times*

```
import string, re
from calibre import strftime
from calibre.web.feeds.recipes import BasicNewsRecipe
from calibre.ebooks.BeautifulSoup import BeautifulSoup

class NYTimes(BasicNewsRecipe):

    title       = 'The New York Times'
    __author__  = 'Kovid Goyal'
    description = 'Daily news from the New York Times'
    timefmt = ' [%a, %d %b, %Y]'
    needs_subscription = True
    remove_tags_before = dict(id='article')
    remove_tags_after  = dict(id='article')
    remove_tags = [dict(attrs={'class':['articleTools', 'post-tools', 'side_tool', 'nextArticleLink c
                dict(id=['footer', 'toolsRight', 'articleInline', 'navigation', 'archive', 'side_sear
                dict(name=['script', 'noscript', 'style'])]
    encoding = 'cp1252'
    no_stylesheets = True
    extra_css = 'h1 {font: sans-serif large;}\n.byline {font:monospace;}'

    def get_browser(self):
        br = BasicNewsRecipe.get_browser()
        if self.username is not None and self.password is not None:
            br.open('http://www.nytimes.com/auth/login')
            br.select_form(name='login')
            br['USERID']   = self.username
            br['PASSWORD'] = self.password
            br.submit()
        return br

    def parse_index(self):
        soup = self.index_to_soup('http://www.nytimes.com/pages/todayspaper/index.html')

        def feed_title(div):
            return ''.join(div.findAll(text=True, recursive=False)).strip()

        articles = {}
        key = None
        ans = []
        for div in soup.findAll(True,
             attrs={'class':['section-headline', 'story', 'story headline']}):

             if div['class'] == 'section-headline':
                 key = string.capwords(feed_title(div))
                 articles[key] = []
                 ans.append(key)

             elif div['class'] in ['story', 'story headline']:
                 a = div.find('a', href=True)
                 if not a:
                     continue
                 url = re.sub(r'\?.*', '', a['href'])
```

```
                url += '?pagewanted=all'
                title = self.tag_to_string(a, use_alt=True).strip()
                description = ''
                pubdate = strftime('%a, %d %b')
                summary = div.find(True, attrs={'class':'summary'})
                if summary:
                    description = self.tag_to_string(summary, use_alt=False)

                feed = key if key is not None else 'Uncategorized'
                if not articles.has_key(feed):
                    articles[feed] = []
                if not 'podcasts' in url:
                    articles[feed].append(
                              dict(title=title, url=url, date=pubdate,
                                   description=description,
                                   content=''))
        ans = self.sort_index_by(ans, {'The Front Page':-1, 'Dining In, Dining Out':1, 'Obituaries':2
        ans = [(key, articles[key]) for key in ans if articles.has_key(key)]
        return ans

    def preprocess_html(self, soup):
        refresh = soup.find('meta', {'http-equiv':'refresh'})
        if refresh is None:
            return soup
        content = refresh.get('content').partition('=')[2]
        raw = self.browser.open('http://www.nytimes.com'+content).read()
        return BeautifulSoup(raw.decode('cp1252', 'replace'))
```

V tomto předpisu vidíme několik nových funkcí. Nejdříve máme:

```
timefmt = ' [%a, %d %b, %Y]'
```

To nastaví zobrazený čas na úvodní stránce vytvořené e-knihy, aby byl ve formátu `Day, Day_Number Month, Year`. Podívejte se na *timefmt* (stránka 51).

Pak vidíme skupinu směrnic pro vyčištění staženého *HTML*:

```
remove_tags_before = dict(name='h1')
remove_tags_after  = dict(id='footer')
remove_tags = ...
```

Tyto odeberou vše před první značkou `<h1>` a vše za první značkou, jejíž id je `footer`. Podívejte se na *remove_tags* (stránka 50), *remove_tags_before* (stránka 51), *remove_tags_after* (stránka 50).

Další zajímavá funkce je:

```
needs_subscription = True
...
def get_browser(self):
    ...
```

`needs_subscription = True` říká Calibre, že tento předpis potřebuje uživatelské jméno a heslo pro přístup k obsahu. To způsobí, že Calibre požádá o uživatelské jméno a heslo, kdykoliv se pokusíte použít tento předpis. Kód v *calibre.web.feeds.news.BasicNewsRecipe.get_browser()* (stránka 44) skutečně provede přihlášení na webovou stránku NYT. Jakmile se přihlásíte, Calibre použije stejnou, přihlášenou instance prohlížeče k načtení celého obsahu. Podívejte se na automatizace[5] k porozumění kódu v `get_browser`.

Další novou funkcí je metoda *calibre.web.feeds.news.BasicNewsRecipe.parse_index()* (stránka 45). Její úlohou je přejít na http://www.nytimes.com/pages/todayspaper/index.html a načíst seznam článků,

[5] http://wwwsearch.sourceforge.net/mechanize/

které se objevují v *dnešních* novinách. I když je to složitější, než jednoduché použití *RSS*, předpis vytvoří e-knihu, která je velice blízká novinám. `parse_index` provádí těžké použití BeautifulSoup[6] pro analýzu webové stránky deníku. Můžete také použít jiné, modernější analyzátory, pokud se vám nelíbí BeatifulSoup. Calibre přichází s lxml[7] a html5lib[8], které jsou doporučenými analyzátory. Pro jejich použití nahraďte volání `index_to_soup()` následujícím:

```
raw = self.index_to_soup(url, raw=True)
# For html5lib
import html5lib
root = html5lib.parse(raw, namespaceHTMLElements=False, treebuilder='lxml')
# For the lxml html 4 parser
from lxml import html
root = html.fromstring(raw)
```

Poslední novou funkcí je metoda *calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()* (stránka 46). Může být použita k provedení libovolných transformací na každé stažené stránce HTML. Zde je použita k obejití reklam, které NYT zobrazuje před každým článkem.

1.2.3 Tipy pro vývoj nových předpisů

Nejlepším způsobem vývoje nových předpisů je použít rozhraní příkazového řádku. Vytvořte předpis pomocí svého oblíbeného editoru Pythonu a uložte ho do souboru `myrecipe.recipe`. Přípona *.recipe* je vyžadována. Pomocí tohoto předpisu můžete stáhnout obsah příkazem:

```
ebook-convert myrecipe.recipe .epub --test -vv --debug-pipeline debug
```

Příkaz **`ebook-convert`** stáhne celé webové stránky a uloží je do souboru EPUB `myrecipe.epub`. `-vv` způsobí, že ebook-convert vyplivne spoustu informací o tom, co dělá. `--test` umožní stáhnout pouze pár článků z nejvýše dvou informačních kanálů. Kromě toho ebook-convert umístí stažené HTML do adresáře `debug/input`, kde `debug` je adresář, který jste zadali ve volbě `--debug-pipeline`.

Jakmile je stahování dokončeno, můžete se podívat na stažené *HTML* otevřením souboru `debug/input/index.html` v prohlížeči. Jakmile jste spokojeni se správností stahování a předzpracování, můžete generovat e-knihy v různých formátech, jak je zobrazeno níže:

```
ebook-convert myrecipe.recipe myrecipe.epub
ebook-convert myrecipe.recipe myrecipe.mobi
...
```

Pokud jste se svým předpisem spokojeni a máte pocit, že je dostatečná poptávka pro ospravedlnění jeho zahrnutí do sady vestavěných předpisů, zveřejněte svůj předpis ve fóru předpisů Calibre[9], abyste ho sdíleli s ostatními uživateli Calibre.

Poznámka: Na OS X jsou nástroje příkazového řádku uvnitř sady calibre, například pokud jste nainstalovali Calibre do `/Applications`, nástroje příkazového řádku jsou v `/Applications/calibre.app/Contents/console.app/Contents/MacOS`

Viz také:

`generated/en/ebook-convert` Rozhraní příkazového řádku pro všechny převody e-knih.

[6] http://www.crummy.com/software/BeautifulSoup/documentation.html
[7] http://lxml.de/
[8] https://github.com/html5lib/html5lib-python
[9] http://www.mobileread.com/forums/forumdisplay.php?f=228

1.2.4 Další čtení

Abyste se dozvěděli více o psaní pokročilých předpisů pomocí některého z vybavení dostupného v `BasicNewsRecipe`, měli byste si pročíst následující zdroje:

API Documentation **(stránka 43)** Dokumentace třídy `BasicNewsRecipe` a všechny její důležité metody a pole.

BasicNewsRecipe[10] Zdrojový kód `BasicNewsRecipe`

Built-in recipes[11] Zdrojový kód vestavěných předpisů, které jsou dodávány s Calibre

The calibre recipes forum[12] Zde se nachází spousta znalostí pisatelů předpisů Calibre.

1.2.5 Dokumentace API

API Documentation for recipes

The API for writing recipes is defined by the *`BasicNewsRecipe`* (stránka 43)

class `calibre.web.feeds.news.`**`BasicNewsRecipe`**(*options*, *log*, *progress_reporter*)

Base class that contains logic needed in all recipes. By overriding progressively more of the functionality in this class, you can make progressively more customized/powerful recipes. For a tutorial introduction to creating recipes, see Přidávání oblíbených webových stránek se zprávami (stránka 23).

`abort_article`(*msg=None*)

Call this method inside any of the preprocess methods to abort the download for the current article. Useful to skip articles that contain inappropriate content, such as pure video articles.

`abort_recipe_processing`(*msg*)

Causes the recipe download system to abort the download of this recipe, displaying a simple feedback message to the user.

`add_toc_thumbnail`(*article*, *src*)

Call this from populate_article_metadata with the src attribute of an <img> tag from the article that is appropriate for use as the thumbnail representing the article in the Table of Contents. Whether the thumbnail is actually used is device dependent (currently only used by the Kindles). Note that the referenced image must be one that was successfully downloaded, otherwise it will be ignored.

classmethod **`adeify_images`**(*soup*)

If your recipe when converted to EPUB has problems with images when viewed in Adobe Digital Editions, call this method from within *`postprocess_html()`* (stránka 46).

`canonicalize_internal_url`(*url*, *is_link=True*)

Return a set of canonical representations of `url`. The default implementation uses just the server hostname and path of the URL, ignoring any query parameters, fragments, etc. The canonical representations must be unique across all URLs for this news source. If they are not, then internal links may be resolved incorrectly.

Parametry **`is_link`** – Is True if the URL is coming from an internal link in an HTML file. False if the URL is the URL used to download an article.

`cleanup`()

Called after all articles have been download. Use it to do any cleanup like logging out of subscription sites, etc.

`clone_browser`(*br*)

Clone the browser br. Cloned browsers are used for multi-threaded downloads, since mechanize is not thread safe. The default cloning routines should capture most browser customization, but if you do something exotic in your recipe, you should override this method in your recipe and clone manually.

Cloned browser instances use the same, thread-safe CookieJar by default, unless you have customized cookie handling.

default_cover(*cover_file*)
Create a generic cover for recipes that don't have a cover

download()
Download and pre-process all articles from the feeds in this recipe. This method should be called only once on a particular Recipe instance. Calling it more than once will lead to undefined behavior. :return: Path to index.html

extract_readable_article(*html*, *url*)
Extracts main article content from 'html', cleans up and returns as a (article_html, extracted_title) tuple. Based on the original readability algorithm by Arc90.

get_article_url(*article*)
Override in a subclass to customize extraction of the *URL* that points to the content for each article. Return the article URL. It is called with *article*, an object representing a parsed article from a feed. See feedparser[13]. By default it looks for the original link (for feeds syndicated via a service like feedburner or pheedo) and if found, returns that or else returns article.link[14].

get_browser(**args*, ***kwargs*)
Return a browser instance used to fetch documents from the web. By default it returns a mechanize[15] browser instance that supports cookies, ignores robots.txt, handles refreshes and has a mozilla firefox user agent.

If your recipe requires that you login first, override this method in your subclass. For example, the following code is used in the New York Times recipe to login for full access:

```
def get_browser(self):
    br = BasicNewsRecipe.get_browser(self)
    if self.username is not None and self.password is not None:
        br.open('http://www.nytimes.com/auth/login')
        br.select_form(name='login')
        br['USERID']   = self.username
        br['PASSWORD'] = self.password
        br.submit()
    return br
```

get_cover_url()
Return a *URL* to the cover image for this issue or *None*. By default it returns the value of the member *self.cover_url* which is normally *None*. If you want your recipe to download a cover for the e-book override this method in your subclass, or set the member variable *self.cover_url* before this method is called.

get_feeds()
Return a list of *RSS* feeds to fetch for this profile. Each element of the list must be a 2-element tuple of the form (title, url). If title is None or an empty string, the title from the feed is used. This method is useful if your recipe needs to do some processing to figure out the list of feeds to download. If so, override in your subclass.

get_masthead_title()
Override in subclass to use something other than the recipe title

get_masthead_url()
Return a *URL* to the masthead image for this issue or *None*. By default it returns the value of the member *self.masthead_url* which is normally *None*. If you want your recipe to download a masthead for the e-book

[13] https://pythonhosted.org/feedparser/
[14] https://pythonhosted.org/feedparser/reference-entry-link.html
[15] http://wwwsearch.sourceforge.net/mechanize/

override this method in your subclass, or set the member variable *self.masthead_url* before this method is called. Masthead images are used in Kindle MOBI files.

get_obfuscated_article(*url*)
If you set *articles_are_obfuscated* this method is called with every article URL. It should return the path to a file on the filesystem that contains the article HTML. That file is processed by the recursive HTML fetching engine, so it can contain links to pages/images on the web.

This method is typically useful for sites that try to make it difficult to access article content automatically.

classmethod image_url_processor(*baseurl*, *url*)
Perform some processing on image urls (perhaps removing size restrictions for dynamically generated images, etc.) and return the precessed URL.

index_to_soup(*url_or_raw*, *raw=False*, *as_tree=False*)
Convenience method that takes an URL to the index page and returns a BeautifulSoup[16] of it.

url_or_raw: Either a URL or the downloaded index page as a string

is_link_wanted(*url*, *tag*)
Return True if the link should be followed or False otherwise. By default, raises NotImplementedError which causes the downloader to ignore it.

Parametry

- **url** – The URL to be followed
- **tag** – The Tag from which the URL was derived

javascript_login(*browser*, *username*, *password*)
This method is used to login to a website that uses javascript for its login form. After the login is complete, the cookies returned from the website are copied to a normal (non-javascript) browser and the download proceeds using those cookies.

An example implementation:

```
def javascript_login(self, browser, username, password):
    browser.visit('http://some-page-that-has-a-login')
    form = browser.select_form(nr=0) # Select the first form on the page
    form['username'] = username
    form['password'] = password
    browser.submit(timeout=120) # Submit the form and wait at most two minutes for loading t
```

Note that you can also select forms with CSS2 selectors, like this:

```
browser.select_form('form#login_form')
browser.select_from('form[name="someform"]')
```

parse_feeds()
Create a list of articles from the list of feeds returned by *BasicNewsRecipe.get_feeds()* (stránka 44). Return a list of `Feed` objects.

parse_index()
This method should be implemented in recipes that parse a website instead of feeds to generate a list of articles. Typical uses are for news sources that have a "Print Edition" webpage that lists all the articles in the current print edition. If this function is implemented, it will be used in preference to *BasicNewsRecipe.parse_feeds()* (stránka 45).

It must return a list. Each element of the list must be a 2-element tuple of the form `('feed title', list of articles)`.

[16] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html

Each list of articles must contain dictionaries of the form:

```
{
'title'       : article title,
'url'         : URL of print version,
'date'        : The publication date of the article as a string,
'description' : A summary of the article
'content'     : The full article (can be an empty string). Obsolete
                do not use, instead save the content to a temporary
                file and pass a file:///path/to/temp/file.html as
                the URL.
}
```

For an example, see the recipe for downloading *The Atlantic*. In addition, you can add 'author' for the author of the article.

If you want to abort processing for some reason and have calibre show the user a simple message instead of an error, call `abort_recipe_processing()` (stránka 43).

populate_article_metadata(*article*, *soup*, *first*)
Called when each HTML page belonging to article is downloaded. Intended to be used to get article metadata like author/summary/etc. from the parsed HTML (soup). :param article: A object of class `calibre.web.feeds.Article`. If you change the summary, remember to also change the text_summary :param soup: Parsed HTML belonging to this article :param first: True iff the parsed HTML is the first page of the article.

postprocess_book(*oeb*, *opts*, *log*)
Run any needed post processing on the parsed downloaded e-book.

Parametry

- **oeb** – An OEBBook object
- **opts** – Conversion options

postprocess_html(*soup*, *first_fetch*)
This method is called with the source of each downloaded *HTML* file, after it is parsed for links and images. It can be used to do arbitrarily powerful post-processing on the *HTML*. It should return *soup* after processing it.

Parametry

- **soup** – A BeautifulSoup[17] instance containing the downloaded *HTML*.
- **first_fetch** – True if this is the first page of an article.

preprocess_html(*soup*)
This method is called with the source of each downloaded *HTML* file, before it is parsed for links and images. It is called after the cleanup as specified by remove_tags etc. It can be used to do arbitrarily powerful pre-processing on the *HTML*. It should return *soup* after processing it.

soup: A BeautifulSoup[18] instance containing the downloaded *HTML*.

preprocess_raw_html(*raw_html*, *url*)
This method is called with the source of each downloaded *HTML* file, before it is parsed into an object tree. raw_html is a unicode string representing the raw HTML downloaded from the web. url is the URL from which the HTML was downloaded.

Note that this method acts *before* preprocess_regexps.

This method must return the processed raw_html as a unicode object.

[17] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
[18] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html

classmethod print_version(*url*)
Take a *url* pointing to the webpage with article content and return the *URL* pointing to the print version of the article. By default does nothing. For example:

```
def print_version(self, url):
    return url + '?&pagewanted=print'
```

skip_ad_pages(*soup*)
This method is called with the source of each downloaded *HTML* file, before any of the cleanup attributes like remove_tags, keep_only_tags are applied. Note that preprocess_regexps will have already been applied. It is meant to allow the recipe to skip ad pages. If the soup represents an ad page, return the HTML of the real page. Otherwise return None.

soup: A BeautifulSoup[19] instance containing the downloaded *HTML*.

sort_index_by(*index*, *weights*)
Convenience method to sort the titles in *index* according to *weights*. *index* is sorted in place. Returns *index*.

index: A list of titles.

weights: A dictionary that maps weights to titles. If any titles in index are not in weights, they are assumed to have a weight of 0.

classmethod tag_to_string(*tag*, *use_alt=True*, *normalize_whitespace=True*)
Convenience method to take a BeautifulSoup[20] *Tag* and extract the text from it recursively, including any CDATA sections and alt tag attributes. Return a possibly empty unicode string.

use_alt: If *True* try to use the alt attribute for tags that don't have any textual content

tag: BeautifulSoup[21] *Tag*

articles_are_obfuscated = False
Set to True and implement *get_obfuscated_article()* (stránka 45) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False
Automatically extract all the text from downloaded article pages. Uses the algorithms from the readability project. Setting this to True, means that you do not have to worry about cleaning up the downloaded HTML manually (though manual cleanup will always be superior).

auto_cleanup_keep = None
Specify elements that the auto cleanup algorithm should never remove. The syntax is a XPath expression. For example:

```
auto_cleanup_keep = '//div[@id="article-image"]' will keep all divs with
                                                  id="article-image"
auto_cleanup_keep = '//*[@class="important"]' will keep all elements
                                                with class="important"
auto_cleanup_keep = '//div[@id="article-image"]|//span[@class="important"]'
                    will keep all divs with id="article-image" and spans
                    with class="important"
```

center_navbar = True
If True the navigation bar is center aligned, otherwise it is left aligned

compress_news_images = False
Set this to False to ignore all scaling and compression parameters and pass images through unmodified. If True and the other compression parameters are left at their default values, jpeg images will be scaled to fit

[19] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
[20] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
[21] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html

in the screen dimensions set by the output profile and compressed to size at most (w * h)/16 where w x h are the scaled image dimensions.

compress_news_images_auto_size = 16
The factor used when auto compressing jpeg images. If set to None, auto compression is disabled. Otherwise, the images will be reduced in size to (w * h)/compress_news_images_auto_size bytes if possible by reducing the quality level, where w x h are the image dimensions in pixels. The minimum jpeg quality will be 5/100 so it is possible this constraint will not be met. This parameter can be overridden by the parameter compress_news_images_max_size which provides a fixed maximum size for images. Note that if you enable scale_news_images_to_device then the image will first be scaled and then its quality lowered until its size is less than (w * h)/factor where w and h are now the *scaled* image dimensions. In other words, this compression happens after scaling.

compress_news_images_max_size = None
Set jpeg quality so images do not exceed the size given (in KBytes). If set, this parameter overrides auto compression via compress_news_images_auto_size. The minimum jpeg quality will be 5/100 so it is possible this constraint will not be met.

conversion_options = {}
Recipe specific options to control the conversion of the downloaded content into an e-book. These will override any user or plugin specified values, so only use if absolutely necessary. For example:

```
conversion_options = {
  'base_font_size'   : 16,
  'tags'             : 'mytag1,mytag2',
  'title'            : 'My Title',
  'linearize_tables' : True,
}
```

cover_margins = (0, 0, '#ffffff')
By default, the cover image returned by get_cover_url() will be used as the cover for the periodical. Overriding this in your recipe instructs calibre to render the downloaded cover into a frame whose width and height are expressed as a percentage of the downloaded cover. cover_margins = (10, 15, '#ffffff') pads the cover with a white margin 10px on the left and right, 15px on the top and bottom. Color names defined at http://www.imagemagick.org/script/color.php Note that for some reason, white does not always work on windows. Use #ffffff instead

delay = 0
Delay between consecutive downloads in seconds. The argument may be a floating point number to indicate a more precise time.

description = u''
A couple of lines that describe the content this recipe downloads. This will be used primarily in a GUI that presents a list of recipes.

encoding = None
Specify an override encoding for sites that have an incorrect charset specification. The most common being specifying `latin1` and using `cp1252`. If None, try to detect the encoding. If it is a callable, the callable is called with two arguments: The recipe object and the source to be decoded. It must return the decoded source.

extra_css = None
Specify any extra *CSS* that should be added to downloaded *HTML* files. It will be inserted into *<style>* tags, just before the closing *</head>* tag thereby overriding all *CSS* except that which is declared using the style attribute on individual *HTML* tags. For example:

```
extra_css = '.heading { font: serif x-large }'
```

feeds = None

List of feeds to download. Can be either `[url1, url2, ...]` or `[('title1', url1), ('title2', url2),...]`

filter_regexps = []
List of regular expressions that determines which links to ignore. If empty it is ignored. Used only if is_link_wanted is not implemented. For example:

```
filter_regexps = [r'ads\.doubleclick\.net']
```

will remove all URLs that have *ads.doubleclick.net* in them.

Only one of *BasicNewsRecipe.match_regexps* (stránka 49) or *BasicNewsRecipe.filter_regexps* (stránka 49) should be defined.

ignore_duplicate_articles = None
Ignore duplicates of articles that are present in more than one section. A duplicate article is an article that has the same title and/or URL. To ignore articles with the same title, set this to:

```
ignore_duplicate_articles = {'title'}
```

To use URLs instead, set it to:

```
ignore_duplicate_articles = {'url'}
```

To match on title or URL, set it to:

```
ignore_duplicate_articles = {'title', 'url'}
```

keep_only_tags = []
Keep only the specified tags and their children. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (stránka 50). If this list is not empty, then the <*body*> tag will be emptied and re-filled with the tags that match the entries in this list. For example:

```
keep_only_tags = [dict(id=['content', 'heading'])]
```

will keep only tags that have an *id* attribute of *"content"* or *"heading"*.

language = 'und'
The language that the news is in. Must be an ISO-639 code either two or three characters long

masthead_url = None
By default, calibre will use a default image for the masthead (Kindle only). Override this in your recipe to provide a url to use as a masthead.

match_regexps = []
List of regular expressions that determines which links to follow. If empty, it is ignored. Used only if is_link_wanted is not implemented. For example:

```
match_regexps = [r'page=[0-9]+']
```

will match all URLs that have *page=some number* in them.

Only one of *BasicNewsRecipe.match_regexps* (stránka 49) or *BasicNewsRecipe.filter_regexps* (stránka 49) should be defined.

max_articles_per_feed = 100
Maximum number of articles to download from each feed. This is primarily useful for feeds that don't have article dates. For most feeds, you should use *BasicNewsRecipe.oldest_article* (stránka 50)

needs_subscription = False
If True the GUI will ask the user for a username and password to use while downloading. If set to "optional" the use of a username and password becomes optional

no_stylesheets = False
Convenient flag to disable loading of stylesheets for websites that have overly complex stylesheets unsuitable for conversion to ebooks formats. If True stylesheets are not downloaded and processed

oldest_article = 7.0
Oldest article to download from this news source. In days.

preprocess_regexps = []
List of *regexp* substitution rules to run on the downloaded *HTML*. Each element of the list should be a two element tuple. The first element of the tuple should be a compiled regular expression and the second a callable that takes a single match object and returns a string to replace the match. For example:

```
preprocess_regexps = [
   (re.compile(r'<!--Article ends here-->.*</body>', re.DOTALL|re.IGNORECASE),
    lambda match: '</body>'),
]
```

will remove everything from *<!–Article ends here–>* to *</body>*.

publication_type = 'unknown'
Publication type Set to newspaper, magazine or blog. If set to None, no publication type metadata will be written to the opf file.

recipe_disabled = None
Set to a non empty string to disable this recipe. The string will be used as the disabled message

recursions = 0
Number of levels of links to follow on article webpages

remove_attributes = []
List of attributes to remove from all tags. For example:

```
remove_attributes = ['style', 'font']
```

remove_empty_feeds = False
If True empty feeds are removed from the output. This option has no effect if parse_index is overridden in the sub class. It is meant only for recipes that return a list of feeds using *feeds* or *get_feeds()* (stránka 44). It is also used if you use the ignore_duplicate_articles option.

remove_javascript = True
Convenient flag to strip all javascript tags from the downloaded HTML

remove_tags = []
List of tags to be removed. Specified tags are removed from downloaded HTML. A tag is specified as a dictionary of the form:

```
{
 name      : 'tag name',   #e.g. 'div'
 attrs     : a dictionary, #e.g. {class: 'advertisment'}
}
```

All keys are optional. For a full explanation of the search criteria, see Beautiful Soup[22] A common example:

```
remove_tags = [dict(name='div', attrs={'class':'advert'})]
```

This will remove all * * tags and all their children from the downloaded *HTML*.

[22] http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html#Searching%20the%20Parse%20Tree

remove_tags_after = None
Remove all tags that occur after the specified tag. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (stránka 50). For example:

```
remove_tags_after = [dict(id='content')]
```

will remove all tags after the first element with *id="content"*.

remove_tags_before = None
Remove all tags that occur before the specified tag. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (stránka 50). For example:

```
remove_tags_before = dict(id='content')
```

will remove all tags before the first element with *id="content"*.

requires_version = (0, 6, 0)
Minimum calibre version needed to use this recipe

resolve_internal_links = False
If set to True then links in downloaded articles that point to other downloaded articles are changed to point to the downloaded copy of the article rather than its original web URL. If you set this to True, you might also need to implement *canonicalize_internal_url()* (stránka 43) to work with the URL scheme of your particular website.

reverse_article_order = False
Reverse the order of articles in each feed

scale_news_images = None
Maximum dimensions (w,h) to scale images to. If scale_news_images_to_device is True this is set to the device screen dimensions set by the output profile unless there is no profile set, in which case it is left at whatever value it has been assigned (default None).

scale_news_images_to_device = True
Rescale images to fit in the device screen dimensions set by the output profile. Ignored if no output profile is set.

simultaneous_downloads = 5
Number of simultaneous downloads. Set to 1 if the server is picky. Automatically reduced to 1 if *BasicNewsRecipe.delay* (stránka 48) > 0

summary_length = 500
Max number of characters in the short description

template_css = u'\n .article_date {\n color: gray; font-family: monospace;\n }\n\n .article_description {\n text-indent: 0
The CSS that is used to style the templates, i.e., the navigation bars and the Tables of Contents. Rather than overriding this variable, you should use *extra_css* in your recipe to customize look and feel.

timefmt = ' [%a, %d %b %Y]'
The format string for the date shown on the first page. By default: Day_Name, Day_Number Month_Name Year

timeout = 120.0
Timeout for fetching files from server in seconds

title = u'Nezn\xe1m\xfd zdroj zpr\xe1v'
The title to use for the ebook

use_embedded_content = None
Normally we try to guess if a feed has full articles embedded in it based on the length of the embedded content. If *None*, then the default guessing is used. If *True* then the we always assume the feeds has embedded content and if *False* we always assume the feed does not have embedded content.

use_javascript_to_login = False
If you set this True, then calibre will use javascript to login to the website. This is needed for some websites that require the use of javascript to login. If you set this to True you must implement the *javascript_login()* (stránka 45) method, to do the actual logging in.

1.3 Prohlížeč e-knih

calibre obsahuje vestavěný prohlížeč e-knih, který umí zobrazit všechny hlavní formáty e-knih. Prohlížeč je vysoce přizpůsobitelný a má mnoho pokročilých funkcí.

- *Spuštění prohlížeče* (stránka 52)
- *Navigace v e-knize* (stránka 52)
- *Přizpůsobení vzhledu a chování vašeho zážitku ze čtení* (stránka 55)
- *Vyhledávání ve slovníku* (stránka 55)
- *Kopírování textu a obrázků* (stránka 55)

1.3.1 Spuštění prohlížeče

Jakoukoliv z knih ve své knihovně calibre můžete zobrazit výběrem knihy a stisknutím tlačítka Zobrazit. To otevře knihu v prohlížeči e-knih. Můžete také spustit samotný prohlížeč z nabídky Start ve Windows nebo pomocí příkazu **ebook-viewer** v Linuxu a OS X (nejdříve musíte nainstalovat nástroje příkazového řádku přejitím na *Předvolby->Rozšířené->Různé*).

1.3.2 Navigace v e-knize

V knize můžete "otáčet stránky" pomocí tlačítek *Další stránka* a *Předchozí stránka* , nebo stisknutím kláves Page Down a Page Up. Na rozdíl od většiny prohlížečů e-knih vás calibre nenutí prohlížet knihy v režimu stránek. Můžete posouvat po množstvích menších než stránka pomocí posuvníku nebo různých přizpůsobitelných klávesových zkratek.

Záložky

Když jste uprostřed knihy a zavřete prohlížeč, zapamatuje si, kde jste přestali číst, a vrátí se tam, když příště otevřete knihu. Můžete také nastavit záložky v knize pomocí tlačítka Záložka . Při prohlížení knih ve formátu EPUB jsou tyto záložky skutečně ukládány do samotného souboru EPUB. Můžete přidat záložky, a pak odeslat soubor přátelům. Když otevřou soubor, uvidí vaše záložky.

Obsah

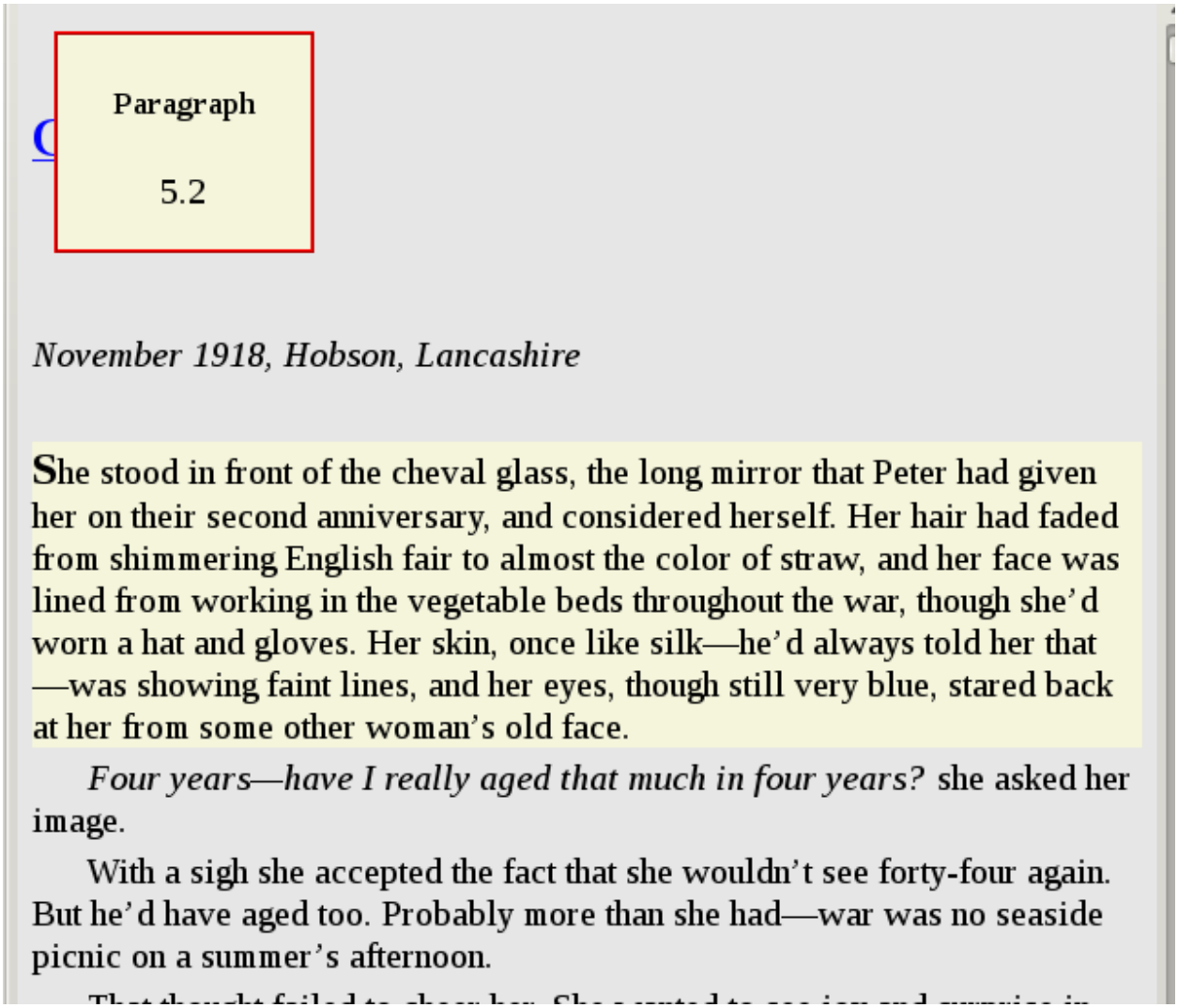
Pokud kniha, kterou čtete, definuje obsah, můžete k němu přistoupit stisknutím tlačítka Obsah . To zobrazí seznam oddílů v knize. Můžete kliknout na kterýkoliv z nich pro přejití na tuto část knihy.

Navigace podle místa

E-knihy, na rozdíl od papírových knih, nemají žádný koncept stránek. Namísto toho když čtete knihu, všimnete si, že je vaše pozice v knize zobrazena v levém horním rohu v poli, jako je toto

. To je jak vaše aktuální pozice, tak celková délka knihy. Tato čísla jsou nezávislá na velikosti obrazovky a písma, ve kterých zobrazujete knihu, a hrají podobnou roli, jako čísla stránek v papírové knize. Můžete zadat, kterékoliv číslo chcete, abyste přešli na odpovídající místo v knize.

calibre má také velice šikovný režim odkazů. Můžete ho zapnout kliknutím na tlačítko Režim odkazů . Jakmile to uděláte, pokaždé, když pohnete myší nad odstavcem, calibre zobrazí jedinečné číslo složené z čísel oddílu a odstavce.


Paragraph
5.2
November 1918, Hobson, Lancashire
She stood in front of the cheval glass, the long mirror that Peter had given her on their second anniversary, and considered herself. Her hair had faded from shimmering English fair to almost the color of straw, and her face was lined from working in the vegetable beds throughout the war, though she'd worn a hat and gloves. Her skin, once like silk—he'd always told her that—was showing faint lines, and her eyes, though still very blue, stared back at her from some other woman's old face.
Four years—have I really aged that much in four years? she asked her image.
With a sigh she accepted the fact that she wouldn't see forty-four again. But he'd have aged too. Probably more than she had—war was no seaside picnic on a summer's afternoon.

Toto číslo můžete použít pro jednoznačné odkazy na části knih, když o nich diskutujete s přáteli, nebo na ně odkazujete v jiných pracích. Tato čísla můžete zadat do pole označeného Přejít na... v horní části okna pro přejití na konkrétní místo odkazu.

Pokud kliknete na odkazy uvnitř e-knihy, aby vás přenesli na různé části knihy, jako je vysvětlivka, můžete použít tlačítka Zpět a Vpřed v levém horním rohu pro návrat tam, kde jste byli. Tato tlačítka se chovají stejně jako ta v internetovém prohlížeči.

1.3.3 Přizpůsobení vzhledu a chování vašeho zážitku ze čtení

Můžete změnit velikosti písem za chodu pomocí tlačítek velikosti písma . Můžete také přepnout prohlížeč na celou obrazovku stisknutím tlačítka Přepnout na celou obrazovku . Kliknutím na tlačítko Předvolby můžete změnit výchozí písma použitá prohlížečem na taková, která se vám líbí, stejně jako výchozí velikost písma po spuštění prohlížeče.

Pokročilejší přizpůsobení může být dosaženo nastavením uživatelské šablony stylů. To je šablona stylů, kterou můžete nastavit, aby se použila na každou knihu. Pomocí ní můžete dělat věci, jako jsou bílý text na černém pozadí, změna stylů odstavce, zarovnání textu do bloku atd. Pro příklady vlastních šablon stylů používaných uživateli calibre navštivte fórum[23].

1.3.4 Vyhledávání ve slovníku

Význam slov v aktuální knize můžete vyhledat označením slova přetažením myší, pak kliknutím pravým tlačítkem myši a výběrem Vyhledat ve slovníku. calibre používá pro vyhledání slov veřejně dostupný server slovníku na `dict.org`. Definice je zobrazena v malém poli ve spodní části obrazovky.

1.3.5 Kopírování textu a obrázků

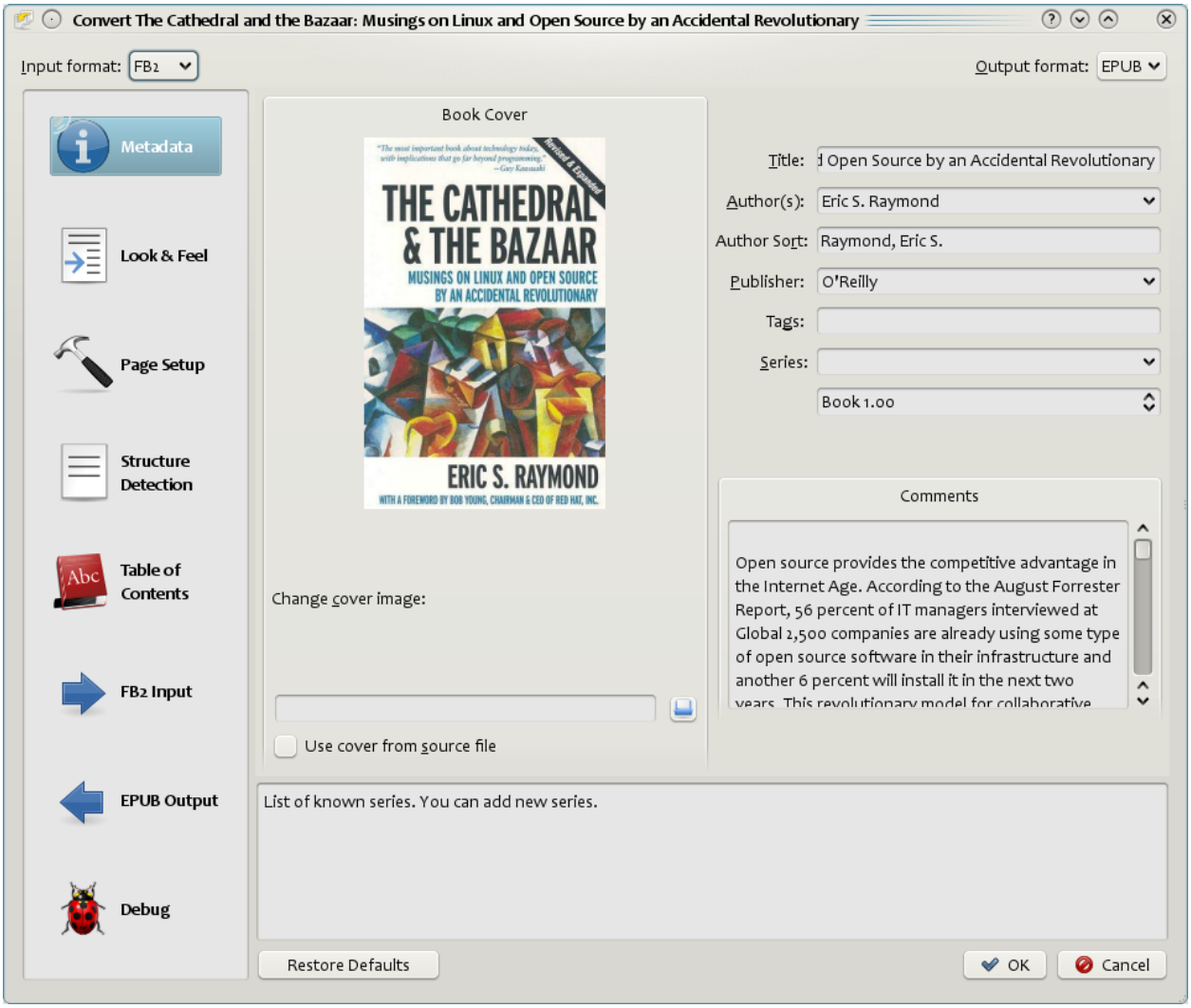
Text a obrázky můžete vybrat přetažením obsahu myší a potom kliknutím pravým tlačítkem myši na Kopírovat. Zkopírovaný materiál může být vložen do jiné aplikace jako prostý text a obrázky.

1.4 Převod e-knihy

Calibre má systém převodu, který je navržen tak, aby se velmi snadno používal. Obyčejně stačí přidat knihu do Calibre, kliknout na převod a Calibre se pokusí vygenerovat výstup, který je co nejpodobnější vstupu. Calibre však přijímá velké množství vstupních formátů, z nichž ne všechny jsou stejně vhodné jako jiné pro převod na e-knihy. V případě takových vstupních formátů, nebo pokud pouze chcete větší kontrolu nad systémem převodu, Calibre má mnoho voleb pro jemné ladění procesu převodu. Pamatujte však, že systému převodu Calibre není náhradou za plnohodnotný editor e-knih. Pro úpravu e-knih je doporučuji nejdříve pomocí Calibre převést do formátu EPUB nebo AZW3, a pak použít funkci Upravit knihu, abyste je dostali do dokonalé podoby. Pak můžete použít upravenou e-knihu jako vstup pro převod do jiných formátů v Calibre.

Tento dokument bude odkazovat hlavně na nastavení převodu, které naleznete v dialogovém okně převodu, jak je zobrazeno níže. Všechna tato nastavení jsou také dostupná prostřednictvím rozhraní příkazového řádku pro převod zdokumentovaného v `generated/en/ebook-convert`. V Calibre můžete získat nápovědu k jakémukoliv individuálnímu nastavení podržením myši nad ním, objeví se popisek tlačítka popisující nastavení.

[23] http://www.mobileread.com/forums/showthread.php?t=51500

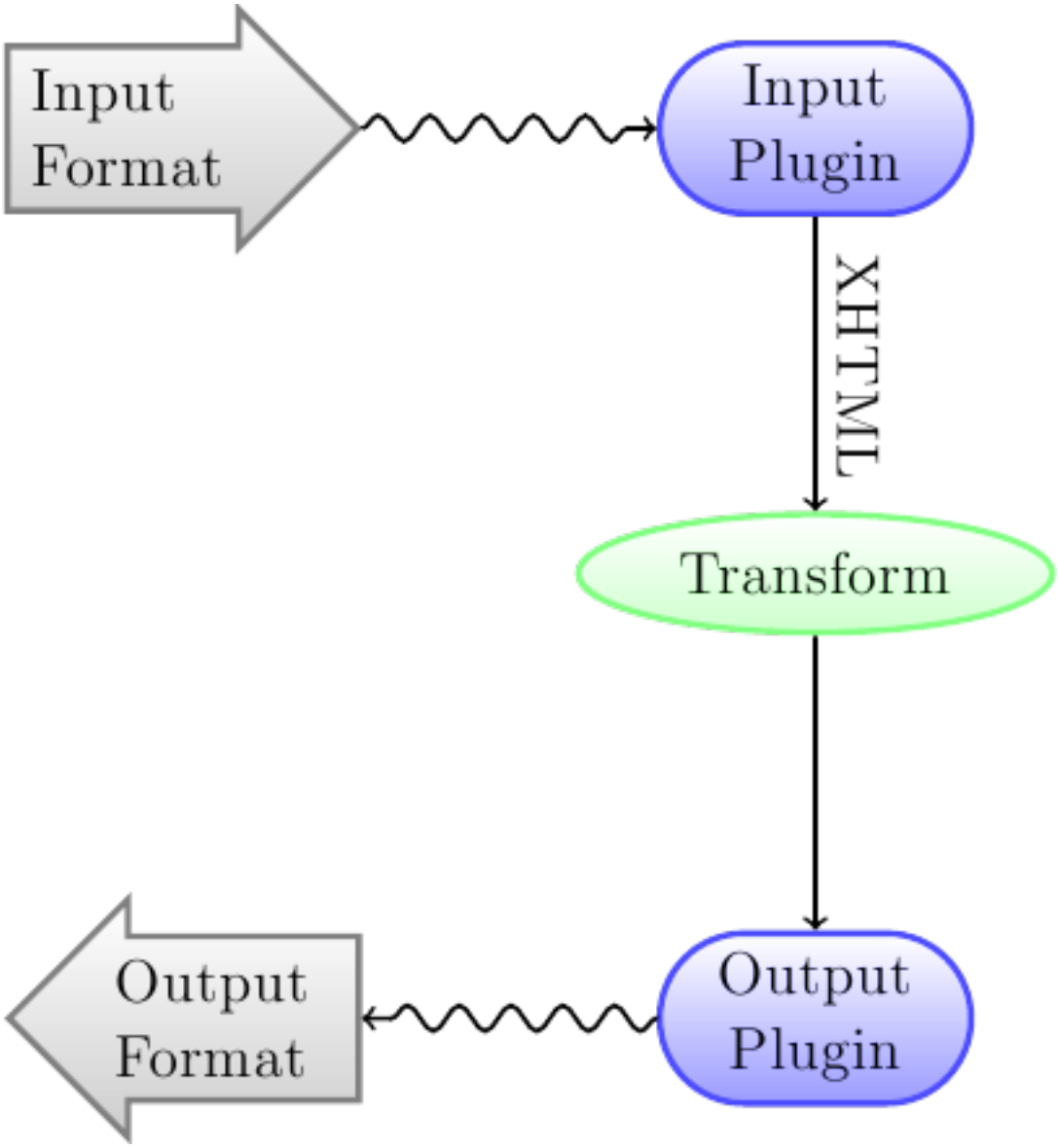

Convert The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary
Input format: FB2
Output format: EPUB
Metadata
Look & Feel
Page Setup
Structure Detection
Table of Contents
FB2 Input
EPUB Output
Debug
Book Cover
THE CATHEDRAL & THE BAZAAR
MUSINGS ON LINUX AND OPEN SOURCE BY AN ACCIDENTAL REVOLUTIONARY
ERIC S. RAYMOND
Change cover image:
Use cover from source file
Title:
Author(s): Eric S. Raymond
Author Sort: Raymond, Eric S.
Publisher: O'Reilly
Tags:
Series:
Book 1.00
Comments
Open source provides the competitive advantage in the Internet Age. According to the August Forrester Report, 56 percent of IT managers interviewed at Global 2,500 companies are already using some type of open source software in their infrastructure and another 6 percent will install it in the next two
List of known series. You can add new series.
Restore Defaults
OK
Cancel

Obsah

- *Úvod* (stránka 56)
- *Vzhled a chování* (stránka 58)
- *Nastavení stránky* (stránka 60)
- *Heuristické zpracování* (stránka 61)
- *Hledat a nahradit* (stránka 62)
- *Rozpoznávání struktury* (stránka 62)
- *Obsah* (stránka 63)
- *Požití obrázku jako názvu kapitol pří konverzi HTML vstupních dokumentů* (stránka 65)
- *Používání atributů značek pro dodání textu pro položky v obsahu* (stránka 65)
- *Jak jsou volby nastaveny nebo uloženy pro převod* (stránka 65)
- *Format specific tips* (stránka 66)

1.4.1 Úvod

První věc, kterou musíte pochopit o systému převodu, je to, že je navržen jako řetězec. Schematicky to vypadá takto:



Vstupní formát se nejdříve převede na XHTML vhodným *vstupním modulem*. Toto HTML se pak *transformuje*. V posledním kroku se zpracované XHTML převede na zadaný výstupní formát vhodným *výstupním modulem*. Výsledky převodu se mohou značně lišit v závislosti na vstupním formátu. Některé formáty se převádějí mnohem lépe než ostatní. Seznam nejlepších zdrojových formátů pro převod je dostupný *zde* (stránka 106).

Transformace, které se provádějí na výstupu XHTML, je místo, kde se odehrává veškerá práce. Existují různé transformace, například pro vložení metadat knihy jako stránky na začátek knihy, pro rozpoznání nadpisů kapitol a automatického vytvoření obsahu, pro proporcionální změnu velikosti písma a tak dále. Je důležité si zapamatovat, že všechny transformace se provádějí na výstupním XHTML pomocí *vstupního modulu*, ne na samotném vstupním souboru. Takže například když požádáte Calibre, aby převedlo soubor RTF na EPUB, tak bude nejdříve vnitřně převeden na XHTML, na toto XHTML budou použity různé transformace, a pak *výstupní modul* vytvoří soubor EPUB, automaticky vygeneruje všechna metadata, obsah a tak dále.

Tento proces můžete vidět v akci použitím volby ladění . Stačí zadat cestu k adresáři pro výstup ladění.

Během převodu Calibre umístí XHTML generované různými fázemi převodního řetězce do různých podadresářů. Čtyři podadresáře jsou:

Tabulka 1.2: Fáze převodního řetězce

Adresář	Popis
vstup	Obsahuje výstup HTML ze vstupního modulu. Použijte pro ladění vstupního modulu.
analyzováno	Výsledek předzpracování a převodu na XHTML výstupu ze vstupního modulu. Použijte pro ladění rozpoznávání struktury.
struktura	Po rozpoznání struktury, ale před vyhlazením CSS a převodem velikosti písma. Použijte pro ladění převodu velikosti písma a transformace CSS.
zpracováno	Těsně před tím, než je e-kniha předána výstupnímu modulu. Použijte pro ladění výstupního modulu.

Pokud chcete trochu upravit vstupní dokument před tím, než ho necháte Calibre převést, to nejlepší, co můžete udělat, je upravit soubory v podadresáři `vstup`, pak je zazipovat a použít soubor zip jako vstupní formát pro následující převody. Abyste to udělali, použijte dialogové okno *Upravit metadata* pro přidání souboru zip jako formátu pro knihu, a pak v levém horním rohu dialogového okna převodu vyberte ZIP jako vstupní formát.

Tento dokument se bude zabývat hlavně různými transformacemi, které pracují na polotovaru XHTML, a jak je ovládat. Na konci je několik konkrétních tipů pro každý vstupní nebo výstupní formát.

1.4.2 Vzhled a chování

Obsah

- *Úprava měřítka velikosti písma* (stránka 58)
- *Mezery mezi odstavci* (stránka 59)
- *Dodatečné CSS* (stránka 60)
- *Různé* (stránka 60)

Tato skupina voleb určuje různé aspekty vzhledu a chování převedené e-knihy.

Úprava měřítka velikosti písma

Jedna z nejhezčích funkcí při čtení e-knih je schopnost snadno upravit velikost písma, aby vyhovovalo jednotlivým potřebám a světelným podmínkám. Calibre má propracované algoritmy, aby zajistilo, že všechny produkované knihy mají jednotné velikosti písma, bez ohledu na to, jaké velikosti písma jsou zadány ve vstupním dokumentu.

Základní velikost písma dokumentu je nejběžnější velikost písma v tomto dokumentu, t. j. velikost větší části textu v tomto dokumentu. Když zadáte *Základní velikost písma*, Calibre automaticky přiměřeně změní měřítko všech velikostí písma v dokumentu, takže nejběžnější velikost písma se stane zadanou základní velikostí písma a ostatním velikostem písma bude přiměřeně upraveno měřítko. Výběrem větší základní velikosti písma můžete písma v dokumentu zvětšit a naopak. Když nastavíte základní velikost písma, měli byste pro dosažení nejlepších výsledků také nastavit klíč velikosti písma.

Obyčejně Calibre automaticky zvolí základní velikost písma vhodnou pro výstupní profil, který jste zvolili (viz *Nastavení stránky* (stránka 60)). Toto však zde můžete přepsat, pokud pro vás výchozí nastavení není vhodné.

Volba *Klíč velikosti písma* vám umožňuje určit, jak se mění nezákladní velikosti písma. Algoritmus změny velikosti písma funguje pomocí klíče velikosti písma, což je jednoduše čárkami oddělený seznam velikostí písma. Klíč velikost písma říká Calibre, o kolik "kroků" větší nebo menší by měla daná velikost písma být v porovnání s základní velikostí písma. Předpokládá se, že by v dokumentu měl být omezený počet velikostí písma. Například jedna velikost pro základní text, několik velikostí pro různé úrovně nadpisů a několik velikostí pro horní nebo dolní indexy a poznámky pod čarou. Klíč velikosti písma umožňuje Calibre rozčlenit velikosti písma ve vstupních dokumentech do samostatných "košů" odpovídajících různým logickým velikostem písem.

Vysvětleme si to na příkladu. Předpokládejme, že zdrojový dokument, který převádíme, byl vytvořen někým s vynikajícím zrakem a má základní velikost písma 8pt. To znamená, že větší část textu v dokumentu má velikost 8pt, zatímco nadpisy jsou o něco větší (řekněme 10 a 12pt) a poznámky pod čarou o něco menší 6pt. Takže pokud použijeme následující nastavení:

```
Base font size : 12pt
Font size key  : 7, 8, 10, 12, 14, 16, 18, 20
```

výstupní dokument bude mít základní velikost písma 12pt, nadpisy 14 a 16pt a poznámky pod čarou 8pt. Nyní předpokládejme, že chceme, aby největší velikost nadpisu více vyčnívala a také aby poznámky pod čarou byly trochu větší. Abychom toho dosáhli, klíč písma by měl být změněn na:

```
New font size key : 7, 9, 12, 14, 18, 20, 22
```

Největší nadpisy budou mít nyní 18pt, zatímco poznámky pod čarou budou mít 9pt. Můžete si hrát s tímto nastavením, abyste se pokusili zjistit, co by pro vás bylo nejlepší pomocí průvodce změny měřítka písma, který můžete otevřít kliknutím na malé tlačítko vedle nastavení *Klíč velikosti písma*.

Může tu být také zakázána veškerá změna velikosti písma v převodu, pokud chcete zachovat velikost písma ve vstupním dokumentu.

Související nastavení je *Výška řádku*. Výška řádku určuje svislou výšku řádků. Ve výchozím nastavení (výška řádku 0) se neprovádí žádná manipulace s výškami řádků. Pokud zadáte nevýchozí hodnotu, budou výšky řádků nastaveny na všech místech, které nemají určenu vlastní výšky řádků. Toto je však taková tupá zbraň a mělo by to být používáno šetrně. Pokud chcete upravit výšky řádků pro nějaký oddíl vstupu, je lepší použít *Dodatečné CSS* (stránka 60).

Mezery mezi odstavci

Obyčejně jsou odstavce v XHTML vykresleny s prázdným řádkem mezi nimi a bez úvodního odsazení textu. Calibre má pár voleb, jak toto určit. *Odebrat mezery mezi odstavci* vynutí zajištění, aby všechny odstavce neměly žádné vnitřní mezery mezi odstavci. Nastaví také odsazení textu na 1,5em (může být změněno) pro označení začátku každého odstavce. *Vložit prázdný řádek* dělá opak, což zaručuje, že je právě jeden prázdný řádek mezi každou dvojicí odstavců. Obě tyto volby jsou velice ucelené, odebírají mezery nebo je vkládají pro *všechny* odstavce (technicky vzato značky <p> a). Je to proto, abyste mohli jen nastavit volbu a být si jistí, že se to provede tak, jak bylo ohlášeno bez ohledu na to, jak je neuspořádaný vstupní soubor. Jedinou výjimkou je případ, kdy vstupní soubor používá pevné konce řádků pro doplnění mezer mezi odstavce.

Pokud chcete odebrat mezery mezi všemi odstavci kromě pár vybraných, nepoužívejte tyto volby. Místo toho přidejte následující kód CSS do *Dodatečné CSS* (stránka 60):

```
p, div { margin: 0pt; border: 0pt; text-indent: 1.5em }
.spacious { margin-bottom: 1em; text-indent: 0pt; }
```

Pak ve zdrojovém dokumentu označte odstavce, které potřebují mezery *class="spacious"*. Pokud váš vstupní dokument není v HTML, použijte volbu ladění popsanou v Úvodu, abyste získali HTML (použijte podadresář `input`).

Dodatečné CSS

Tato volba vám umožňuje zadat libovolné CSS, které bude použito na všechny soubory HTML na vstupu. Toto CSS je použito s velice vysokou prioritou, a proto by mělo přepsat většinu CSS přítomného v samotném **vstupním dokumentu**. Toto nastavení můžete použít pro jemné ladění prezentace nebo rozložení dokumentu. Například pokud chcete, aby všechny odstavce třídy *endnote* byly zarovnány vpravo, stačí přidat:

```
.endnote { text-align: right }
```

nebo pokud chcete změnit odsazení všech odstavců:

```
p { text-indent: 5mm; }
```

Dodatečné CSS je velice výkonná volba, ale potřebujete pochopit, jak CSS funguje, pro použití jejího plného potenciálu. Můžete použít volbu ladění řetězce popsanou výše, abyste viděli, jaké CSS je přítomné ve vstupním dokumentu.

Různé

V tomto oddílu je několik dalších voleb.

No text justification Obyčejně, pokud to výstupní formát podporuje, Calibre vynutí, aby výstupní e-kniha měla *zarovnaný* text (tj. hladký pravý okraj). Tato volba vypne toto chování, v takovém případě bez ohledu na zarovnání zadané ve vstupním dokumentu bude použito namísto toho.

Linearize tables Některé chybně navržené dokumenty používají tabulky pro ovládání rozložení textu na stránce. Při převodu mají tyto dokumenty často text, který vybíhá mimo stránku a jiné artefakty. Tato volba extrahuje obsah z tabulek a zobrazí ho v lineárním tvaru. Pamatujte, že tato volba převede na řádky *všechny* tabulky, takže ji používejte, pouze pokud jste si jisti, že vstupní dokument nepoužívá tabulky pro oprávněné účely, jako je zobrazení tabulkových informací.

Transliterate unicode characters Překódovat znaky unicode do zastoupení ASCII. Používejte opatrně, protože to nahradí znaky unicode za ASCII. Například to nahradí "Михаил Горбачёв" za "Michail Gorbačov". Také pamatujte, že v případech, kdy je více zastoupení znaků (například znaky společné pro čínštinu a japonštinu), bude použito zastoupení používané největším počtem lidí (například čínština v předchozím). Tato volba je hlavně užitečná, pokud se chystáte zobrazit e-knihu na zařízení, které nemá podporu unicode.

Input character encoding Starší dokumenty někdy nezadávají své kódování znaků. Při převodu to může vést k neanglické znakům nebo speciálním znakům, jako je poškození chytrých uvozovek. Calibre zkouší automaticky rozpoznat kódování znaků zdrojového dokumentu, ale není vždy úspěšné. Pomocí tohoto nastavení můžete vynutit předpokládání konkrétního kódování znaků. *cp1252* je běžné kódování pro dokumenty vytvořené pomocí softwaru Windows. Měli byste si také přečíst *How do I convert my file containing non-English characters, or smart quotes?* (stránka 106) pro více informací o problematice kódování.

1.4.3 Nastavení stránky

Volby Nastavení stránky jsou pro ovládání rozložení obrazovky, jako jsou okraje a velikosti obrazovky. Existují volby pro nastavení okrajů stránky, které budou použity Výstupním modulem, pokud vybraný Výstupní formát podporuje okraje stránky. Kromě toho byste měli vybrat Vstupní profil a Výstupní profil. Obě sady profilů se v podstatě zabývají tím, jak interpretovat rozměry ve vstupních a výstupních dokumentech, velikosti obrazovky a klíče úpravy měřítka výchozího písma.

Pokud víte, že soubor, který převádíte, byl zamýšlen k použití na konkrétním zařízení nebo softwarové platformě, zvolte odpovídající vstupní profil, jinak prostě zvolte výchozí vstupní profil. Pokud víte, že soubory, které vytváříte, jsou určeny pro konkrétní typ zařízení, zvolte odpovídající výstupní profil. Zejména pro výstupní soubory MOBI byste měli zvolit Kindle, pro LIT čtečku Microsoft a pro EPUB čtečku Sony. V případě EPUB bude mít profil čtečky Sony za výsledek soubory EPUB, které budou fungovat všude. Má to však některé vedlejší účinky, jako je umělé vkládání

konců oddílů pro zachování vnitřních součástí pod mezní hodnotou velikosti, což je vyžadování pro zařízení SONY. Zvláště pro telefony iPhone nebo Android zvolte vyberte výstupní profil SONY. Pokud víte, že vaše soubory EPUB nebudou čteny na SONY nebo podobném zařízení, použijte výchozí výstupní profil. Pokud vytváříte soubory MOBI, které nejsou zamýšleny pro Kindle, zvolte výstupní profil knih Mobipocket.

Výstupní profil také určuje velikost obrazovky. To například způsobí, že v některých výstupních formátech bude u obrázků automaticky změněna velikost, aby se přizpůsobily obrazovce. Takže zvolte profil zařízení, které má podobnou velikost obrazovky vašemu zařízení.

1.4.4 Heuristické zpracování

Heuristické zpracování poskytuje celou řadu funkcí, které mohou být použity pro pokus o rozpoznání a opravu běžných problémů ve špatně formátovaných vstupních dokumentech. Použijte tyto funkce, pokud váš vstupní dokument trpí špatným formátování Protože tyto funkce spoléhají na běžné vzory, uvědomte si, že v některých případech může volba vést k horším výsledkům, tak to používejte opatrně. Například několik z těchto voleb odebere všechny entity pevných mezer, nebo mohou zahrnovat falešně pozitivní shody souvisejících s funkcí.

Enable heuristic processing Tato volba aktivuje fázi heuristického zpracování převodního řetězce Calibre. Musí to být povoleno, aby byly použity různé dílčí funkce.

Unwrap lines Povolení této volby způsobí, že se Calibre pokusí rozpoznat a opravit tvrdé pevné konce řádků existujících v dokumentu pomocí interpunkčních vodítek a délky řádku. Calibre se nejdříve pokusí rozpoznat, zda pevné konce řádků existují, pokud se zdá, že neexistují, Calibre se nepokusí zrušit zalomení řádků. Koeficient zrušení zalomení řádku může být zmenšen, pokud chcete 'vynutit', aby Calibre zrušilo zalomení řádků.

Line-unwrap factor Tato volba určuje algoritmus, který Calibre používá k odebrání pevných konců řádků. Například pokud je hodnota této volby 0,4, znamená to, že Calibre odebere pevné konce řádků z konců řádků, jejichž délky jsou menší než délka 40 % všech řádků v dokumentu. Pokud má váš dokument pouze několik konců řádků, které potřebují opravu, pak by tato hodnota měla být snížena někam mezi 0,1 a 0,2.

Detect and markup unformatted chapter headings and sub headings Pokud váš dokument neobsahuje nadpisy kapitol a názvy formátované odlišně od zbytku textu, Calibre může použít tuto volbu, aby se je pokusilo rozpoznat a obklopit je značkami nadpisů. Značky <h2> se používají pro nadpisy kapitol; značky <h3> se používají pro jakékoliv rozpoznané názvy.

Tato funkce nevytvoří obsah, ale v mnoha případech způsobí, že výchozí nastavení rozpoznání kapitol Calibre správně rozpozná kapitoly a sestaví obsah. Upravte XPath v Rozpoznávání struktury, pokud není obsah automaticky vytvořen. Pokud v dokumentu nejsou použity žádné další nadpisy, pak nastavení "//h:h2" v Rozpoznávání struktury bude nejsnadnější způsob, jak vytvořit obsah pro dokument.

Tyto vložené nadpisy nejsou naformátovány, pro použití formátování použijte volbu Dodatečné CSS pod Vzhledem a chováním v nastavení převodu. Například pro značky zarovnání nadpisů na střed použijte následující:

```
h2, h3 { text-align: center }
```

Renumber sequences of <h1> or <h2> tags Někteří vydavatelé formátují nadpisy kapitol pomocí více značek <h1> nebo <h2> postupně. Výchozí nastavení převodu Calibre způsobí, že takové nadpisy budou rozděleny na dvě části. Tato volba přečísluje značky nadpisů, aby se zabránilo rozdělení.

Delete blank lines between paragraphs Tato volba způsobí, že Calibre analyzuje prázdné řádky obsažené v dokumentu. Pokud je každý odstavec proložený prázdným řádkem, pak Calibre odebere všechny tyto prázdné odstavce. Sekvence více prázdných řádků budou považovány za přerušení příběhu a budou zachovány jako jeden odstavec. Tato volba se liší od volby 'Odebrat mezery mezi odstavci' pod 'Vzhled a chování' v tom, že skutečně mění obsah HTML, zatímco druhá volba mění styly dokumentu. Tato volba může také odebrat odstavce, které byly vloženy pomocí volby Calibre 'Vložit prázdný řádek'.

Ensure scene breaks are consistently formatted Pomocí této volby se Calibre pokusí rozpoznat běžné značky přerušení příběhu a zajistit, aby byly zarovnané na střed. Značkám 'měkkého' přerušení příběhu, tj. přerušení příběhu

definované pouze dodatečnými prázdnými znaky, jsou změněny styly, aby se zajistilo, že nebudou zobrazeny ve spojení s konci stránek.

Replace scene breaks Pokud je tato volba konfigurována, pak Calibre nahradí značky přerušení příběhu, které najde, nahrazovacím textem zadaným uživatelem. Pamatujte, prosím, že některé ozdobné znaky nemusí být podporovány na všech čtecích zařízení.

Obecně byste se měli vyhnout používání značek HTML, Calibre zahodí jakékoliv značky a použije předem definované značení. Značky <hr />, tj. vodorovné čáry, a značky <img> jsou výjimky. Vodorovné čáry mohou být volitelně zadány styly, pokud se rozhodnete přidat vlastní styl, nezapomeňte zahrnout nastavení šířky 'width', jinak bude informace stylu zahozena. Můžete použít značky obrázků, ale Calibre neposkytuje možnost přidat obrázek během převodu, to musí být provedeno po faktickém použití funkce 'Upravit knihu'.

Příklad značky obrázku (po převodu umístí obrázek do složky 'Images' uvnitř souboru epub):

Příklad vodorovné čáry se styly: <hr style="width:20%;padding-top: 1px;border-top: 2px ridge black;border-bottom: 2px groove black;"/>

Remove unnecessary hyphens Calibre analyzuje veškerý obsah rozdělený spojovníky v dokumentu, když je tato volba povolena. Samotný dokument je použit jako slovník pro analýzu. To umožňuje Calibre přesně odebrat spojovníky pro jakákoliv slova v dokumentu v jakémkoliv jazyce, společně s vymyšlenými a neznámými vědeckými slovy. Primární nevýhodou je, že slova objevující se v dokumentu pouze jednou nebudou změněna. Analýza probíhá ve dvou průchodech, první průchod analyzuje konce řádků. Řádkům je zrušeno zalomení, pouze pokud v dokumentu slovo existuje s nebo bez spojovníku. Druhý průchod analyzuje všechna slova rozdělená spojovníky v celém dokumentu, spojovníky jsou odebrány, pokud slovo existuje někde jinde v dokumentu bez shody.

Italicize common words and patterns Pokud je povoleno, bude Calibre hledat obecná slova a vzory, které označují kurzívu a převede je na kurzívu. Příkladem jsou obecné textové zásady, jako je ~slovo~, nebo fráze, které by obecně měly být kurzívou, např. latinské fráze, jako jsou 'etc.' nebo 'et cetera'.

Replace entity indents with CSS indents Některé dokumenty používají zásadu definování odsazení textu pomocí entit pevných mezer. Když je tato volba povolena, Calibre se pokusí rozpoznat tento druh formátování a převést je na 3% odsazení textu pomocí CSS.

1.4.5 Hledat a nahradit

Tyto volby jsou užitečné především pro převody dokumentů PDF nebo převody OCR, i když mohou být také použity pro opravu mnoha konkrétních problémů dokumentů. Například některé převody mohou zanechat záhlaví a zápatí stránek v textu. Tyto volby používají regulární výrazy, aby se pokusily rozpoznat záhlaví, zápatí nebo jiný libovolný text a odebrat ho nebo nahradit. Nezapomeňte, že pracují s polotovarem XHTML vytvořeným převodním řetězcem. K dispozici je průvodce, aby vám pomohl přizpůsobit regulární výrazy pro váš dokument. Klikněte na kouzelnou hůlku vedle pole výrazu a klikněte na tlačítko 'Tesovat' po sestavení vyhledávacího výrazu. Úspěšné shody budou zvýrazněny žlutě.

Vyhledávání funguje pomocí regulárního výrazu Pythonu. Všechny odpovídající texty jsou jednoduše odebrány z dokumentu nebo nahrazeny pomocí vzoru nahrazení. Vzor nahrazení je nepovinný, pokud zůstane prázdný, pak text odpovídající hledanému vzoru bude z dokumentu odstraněn. Další informace o regulárních výrazech a jejich syntaxi můžete zjistit v *Vše o používání regulárních výrazů v Calibre* (stránka 163).

1.4.6 Rozpoznávání struktury

Rozpoznávání struktury zahrnuje to, že se Calibre pokouší co nejlépe rozpoznat strukturální prvky ve vstupním dokumentu, pokud nejsou správně zadány. Například kapitoly, konce stránek, záhlaví, zápatí atd. Jak si dokážete představit,

tento proces se velice liší knihu od knihy. Naštěstí má Calibre velice mocné možnosti, aby toto určilo. S mocí přichází složitost, ale pokud si jednou uděláte čas na to naučit se to složité, zjistíte, že to stálo za námahu.

Konce kapitol a stránek

Calibre má dvě sady možností pro *rozpoznávání kapitol* a *vkládání konců stránek*. To může být někdy trochu matoucí, protože ve výchozím nastavení Calibre vloží konce stránek před rozpoznané kapitoly, stejně jako před místa rozpoznaná možností konců stránek. Důvodem pro to je to, že často existují místa, kam by měly být vloženy konce stránek, která nejsou hranicemi kapitoly. Rozpoznané kapitoly mohou být také volitelně vloženy do automaticky generovaného obsahu.

Calibre používá *XPath*, výkonný jazyk, který umožňuje uživateli zadat hranice kapitol nebo konce stránek. XPath se může zdát nejdříve trochu skličující pro používání, naštěstí je zde *kurz XPath* (stránka 131) v uživatelské příručce. Nezapomeňte, že Rozpoznávání struktury pracuje na polotovaru XHTML vytvořeného převodním řetězcem. Použijte možnost ladění popsanou v *Úvod* (stránka 56), abyste určili vhodné nastavení pro svou knihu. Je zde také tlačítko pro Průvodce XPath, aby pomohl s generováním jednoduchých výrazů XPath.

Ve výchozím nastavení používá Calibre následující výraz pro rozpoznávání kapitol:

```
//*[((name()='h1' or name()='h2') and re:test(., 'chapter|book|section|part\s+', 'i')) or @class = 'c
```

Tento výraz je poměrně složitý, protože se snaží zpracovat velký počet běžných případů souběžně. To znamená, že Calibre bude předpokládat, že kapitoly začínají buď značkou <*h1*> nebo <*h2*>, které obsahují kterékoliv z těchto slov *(kapitola, kniha, oddíl nebo část)*, nebo které mají atribut *class="kapitola"*.

Související možností je *Označení kapitol*, který vám umožní určit, co Calibre udělá, když rozpozná kapitolu. Ve výchozím nastavení vloží konec stránky před kapitolu. Můžete ho nechat vložit linku namísto nebo navíc ke konci stránky. Můžete ho také nechat nedělat nic.

Výchozí nastavení pro rozpoznávání konců stránek je:

```
//*[name()='h1' or name()='h2']
```

což znamená, že Calibre ve výchozím nastavení vloží konce stránek před každou značku <*h1*> a <*h2*>.

Poznámka: Výchozí výrazy se mohou měnit v závislosti na vstupním formátu, který převádíte.

Různé

V tomto oddílu je několik dalších voleb.

Insert metadata as page at start of book Jednou ze skvělých věcí na Calibre je to, že vám umožňuje udržovat velice kompletní metadata o všech vašich knihách, například hodnocení, štítky, komentáře atd. Tato možnost vytvoří jednoduchou stránku se všemi těmito metadaty a vloží ji do převedené e-knihy, obvykle hned za obálku. Berte to jako způsob, jak si vytvořit svůj vlastní přizpůsobený přebal knihy.

Remove first image Někdy zdrojový dokument, který převádíte, obsahuje obálku jako součást knihy namísto samostatné obálky. Pokud jste také zadali obálku v Calibre, pak bude mít převedená kniha dvě obálky. Tato možnost jednoduše odebere první obrázek ze zdrojového dokumentu, čímž zajistí, že převedená kniha má pouze jednu obálku, tu zadanou v Calibre.

1.4.7 Obsah

Když má vstupní dokument ve svých metadatech obsah, Calibre ho prostě použije. Nicméně velké množství starších formátů buď nepodporuje obsah založený na metadatech nebo ho jednotlivé dokumenty nemají. V těchto případech

vám volby v tomto oddílu mohou pomoct automaticky vygenerovat obsah v převáděné e-knize založený na skutečném obsahu ve vstupním dokumentu.

Poznámka: Používání těchto voleb může být trochu složitější, abyste získali přesně to pravé. Pokud upřednostňujete vytváření nebo úpravu obsahu ručně, převeďte na formáty EPUB nebo AZW3 a vyberte zaškrtávací políčko v dolní oddílu Obsah v dialogovém okně převodu, která říká: *Ručně jemně ladit obsah po dokončení převodu*. Tímto se po převodu spustí nástroj Upravit obsah. Umožní vám vytvářet položky v obsahu jednoduše kliknutím na místo v knize, kam chcete, aby položka ukazovala. Můžete také použít samotný nástroj Upravit obsah, aniž byste prováděli převod. Přejděte na *Předvolby->Panely nástrojů* a přidejte Upravit obsah na hlavní panel nástrojů. Pak stačí vybrat knihu, kterou chcete upravit, a kliknout na tlačítko Upravit obsah.

První volba je *Vynutit použití automaticky generovaného obsahu*. Zaškrtnutím této volby můžete nechat Calibre přepsat jakýkoliv nalezený obsah v metadatech vstupního dokumentu automaticky vygenerovaným.

Výchozí způsob, jakým vytvoření automaticky vygenerovaného obsahu funguje, je ten, že Calibre nejdříve zkusí přidat všechny rozpoznané kapitoly do vygenerovaného obsahu. Jak přizpůsobit rozpoznávání kapitol můžete zjistit ve výše uvedeném oddílu *Rozpoznávání struktury* (stránka 62). Pokud nechcete zahrnout rozpoznané kapitoly do vygenerovaného obsahu, zaškrtněte volbu *Nepřidávat rozpoznané kapitoly do obsahu*.

Pokud bylo rozpoznáno méně kapitol, než je hodnota *Mezní hodnota kapitol*, Calibre pak přidá do obsahu všechny hypertextové odkazy, které najde ve vstupním dokumentu. Toto často funguje dobře na mnoha vstupních dokumentech, které obsahují hypertextové odkazy obsahu hned na začátku. Volba *Počet odkazů* může být použita k ovládání tohoto chování. Pokud je nastavena na nulu, nejsou přidány žádné odkazy. Pokud je nastavena na číslo větší než nula, je přidán nanejvýše tento počet odkazů.

Calibre automaticky vyfiltruje duplicity z generovaného obsahu. Pokud však jsou zde nějaké další nežádoucí položky, můžete je vyfiltrovat pomocí volby *Filtr obsahu*. To je regulární výraz, který porovná názvy položek ve vygenerovaném obsahu. Kdykoliv je nalezena shoda, bude odebrána. Například pro odebrání všech názvů položek “Další” nebo “Předchozí” použijte:

```
Next|Previous
```

Volby *Obsah 1., 2. a 3. úrovně* vám umožní vytvořit propracovaný víceúrovňový obsah. Jsou to výrazy XPath, které porovnají značky v polotovaru XHTML vytvořeného převodním řetězcem. Podívejte se na *Úvod* (stránka 56), jak získat přístup k tomuto XHTML. Přečtěte si také *Kurz XPath* (stránka 131), abyste zjistili, jak vytvářet výrazy XPath. Vedle každé volby je tlačítko, které spustí průvodce, který vám pomůže s vytvořením základních výrazů XPath. Následující jednoduchý příklad vysvětluje, jak používat tyto volby.

Předpokládejte, že máte vstupní dokument, jehož výstupem je XHTML, které vypadá takto:

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Sample document</title>
    </head>
    <body>
        <h1>Chapter 1</h1>
        ...
        <h2>Section 1.1</h2>
        ...
        <h2>Section 1.2</h2>
        ...
        <h1>Chapter 2</h1>
        ...
        <h2>Section 2.1</h2>
        ...
    </body>
</html>
```

Potom, nastavíme možnosti jako:

```
Level 1 TOC : //h:h1
Level 2 TOC : //h:h2
```

Toto bude mít za výsledek automaticky generovaný dvouúrovňový obsah, který vypadá jako:

```
Chapter 1
    Section 1.1
    Section 1.2
Chapter 2
    Section 2.1
```

Varování: Ne všechny formáty podporují víceúrovňové obsahy (Table of Contents). Jako první výstupní formát zkuste EPUB. Pokud ten funguje, skuste formát vlastního výběru.

1.4.8 Požití obrázku jako názvu kapitol pří konverzi HTML vstupních dokumentů

Předpokládejme, že chcete použít obrázek jako název kapitoly, ale stále chcete, aby Calibre bylo pro vás schopno automaticky generovat obsah z názvů kapitol. Použijte následující značení HTML, abyste toho dosáhli

```
<html>
    <body>
        <h2>Chapter 1</h2>
        <p>chapter 1 text...</p>
        <h2 title="Chapter 2"><img src="chapter2.jpg" /></h2>
        <p>chapter 2 text...</p>
    </body>
</html>
```

Nastavte *Obsah 1. úrovně* na `//h:h2`. Pak pro druhou kapitolu Calibre vezme název z hodnoty atributu `title` značky `<h2>`, protože značka nemá žádný text.

1.4.9 Používání atributů značek pro dodání textu pro položky v obsahu

Pokud máte obzvlášť dlouhé názvy kapitol a chcete mít v obsahu zkrácené verze, můžete použít atribut názvu, abyste toho dosáhli, například:

```
<html>
    <body>
        <h2 title="Chapter 1">Chapter 1: Some very long title</h2>
        <p>chapter 1 text...</p>
        <h2 title="Chapter 2">Chapter 2: Some other very long title</h2>
        <p>chapter 2 text...</p>
    </body>
</html>
```

Nastavte *Obsah 1. úrovně* na `//h:h2/@title`. Pak Calibre vezme názvy z hodnot atributů `title` značek `<h2>`, namísto použití textu uvnitř značky. Pamatujte na koncové `/@title` ve výrazu XPath, tento formulář můžete použít, abyste řekli Calibre, aby získalo text z jakéhokoliv atributu chcete.

1.4.10 Jak jsou volby nastaveny nebo uloženy pro převod

There are two places where conversion options can be set in calibre. The first is in Preferences->Conversion. These settings are the defaults for the conversion options. Whenever you try to convert a new book, the settings set here will

be used by default.

You can also change settings in the conversion dialog for each book conversion. When you convert a book, calibre remembers the settings you used for that book, so that if you convert it again, the saved settings for the individual book will take precedence over the defaults set in Preferences. You can restore the individual settings to defaults by using the Restore to defaults button in the individual book conversion dialog. You can remove the saved settings for a group of books by selecting all the books and then clicking the edit metadata button to bring up the bulk metadata edit dialog, near the bottom of the dialog is an option to remove stored conversion settings.

When you Bulk Convert a set of books, settings are taken in the following order (last one wins):

- From the defaults set in Preferences->Conversion
- From the saved conversion settings for each book being converted (if any). This can be turned off by the option in the top left corner of the Bulk Conversion dialog.
- From the settings set in the Bulk conversion dialog

Note that the final settings for each book in a Bulk Conversion will be saved and re-used if the book is converted again. Since the highest priority in Bulk Conversion is given to the settings in the Bulk Conversion dialog, these will override any book specific settings. So you should only bulk convert books together that need similar settings. The exceptions are metadata and input format specific settings. Since the Bulk Conversion dialog does not have settings for these two categories, they will be taken from book specific settings (if any) or the defaults.

Poznámka: You can see the actual settings used during any conversion by clicking the rotating icon in the lower right corner and then double clicking the individual conversion job. This will bring up a conversion log that will contain the actual settings used, near the top.

1.4.11 Format specific tips

Here you will find tips specific to the conversion of particular formats. Options specific to particular format, whether input or output are available in the conversion dialog under their own section, for example *TXT Input* or *EPUB Output*.

Convert Microsoft Word documents

calibre can automatically convert `.docx` files created by Microsoft Word 2007 and newer. Just add the file to calibre and click convert (make sure you are running the latest version of calibre as support for `.docx` files is very new).

Poznámka: There is a demo .docx file[24] that demonstrates the capabilities of the calibre conversion engine. Just download it and convert it to EPUB or AZW3 to see what calibre can do.

---

calibre will automatically generate a Table of Contents based on headings if you mark your headings with the `Heading 1`, `Heading 2`, etc. styles in Word. Open the output ebook in the calibre viewer and click the Table of Contents button to view the generated Table of Contents.

Older .doc files

For older .doc files, you can save the document as HTML with Microsoft Word and then convert the resulting HTML file with calibre. When saving as HTML, be sure to use the "Save as Web Page, Filtered" option as this will produce clean HTML that will convert well. Note that Word produces really messy HTML, converting it can take a long time, so be patient. If you have a newer version of Word available, you can directly save it as docx as well.

[24] http://calibre-ebook.com/downloads/demos/demo.docx

Another alternative is to use the free OpenOffice. Open your .doc file in OpenOffice and save it in OpenOffice's format .odt. calibre can directly convert .odt files.

Convert TXT documents

TXT documents have no well defined way to specify formatting like bold, italics, etc, or document structure like paragraphs, headings, sections and so on, but there are a variety of conventions commonly used. By default calibre attempts automatic detection of the correct formatting and markup based on those conventions.

TXT input supports a number of options to differentiate how paragraphs are detected.

Paragraph Style: Auto Analyzes the text file and attempts to automatically determine how paragraphs are defined. This option will generally work fine, if you achieve undesirable results try one of the manual options.

Paragraph Style: Block Assumes one or more blank lines are a paragraph boundary:

```
This is the first.

This is the
second paragraph.
```

Paragraph Style: Single Assumes that every line is a paragraph:

```
This is the first.
This is the second.
This is the third.
```

Paragraph Style: Print Assumes that every paragraph starts with an indent (either a tab or 2+ spaces). Paragraphs end when the next line that starts with an indent is reached:

```
  This is the
first.
  This is the second.

  This is the
third.
```

Paragraph Style: Unformatted Assumes that the document has no formatting, but does use hard line breaks. Punctuation and median line length are used to attempt to re-create paragraphs.

Formatting Style: Auto Attempts to detect the type of formatting markup being used. If no markup is used then heuristic formatting will be applied.

Formatting Style: Heuristic Analyzes the document for common chapter headings, scene breaks, and italicized words and applies the appropriate html markup during conversion.

Formatting Style: Markdown calibre also supports running TXT input though a transformation preprocessor known as markdown. Markdown allows for basic formatting to be added to TXT documents, such as bold, italics, section headings, tables, lists, a Table of Contents, etc. Marking chapter headings with a leading # and setting the chapter XPath detection expression to "//h:h1" is the easiest way to have a proper table of contents generated from a TXT document. You can learn more about the markdown syntax at daringfireball[25].

Formatting Style: None Applies no special formatting to the text, the document is converted to html with no other changes.

[25] http://daringfireball.net/projects/markdown/syntax

Convert PDF documents

PDF documents are one of the worst formats to convert from. They are a fixed page size and text placement format. Meaning, it is very difficult to determine where one paragraph ends and another begins. calibre will try to unwrap paragraphs using a configurable, *Line Un-Wrapping Factor*. This is a scale used to determine the length at which a line should be unwrapped. Valid values are a decimal between 0 and 1. The default is 0.45, just under the median line length. Lower this value to include more text in the unwrapping. Increase to include less. You can adjust this value in the conversion settings under *PDF Input*.

Also, they often have headers and footers as part of the document that will become included with the text. Use the Search and Replace panel to remove headers and footers to mitigate this issue. If the headers and footers are not removed from the text it can throw off the paragraph unwrapping. To learn how to use the header and footer removal options, read *Vše o používání regulárních výrazů v Calibre* (stránka 163).

Some limitations of PDF input are:

- Complex, multi-column, and image based documents are not supported.
- Extraction of vector images and tables from within the document is also not supported.
- Some PDFs use special glyphs to represent ll or ff or fi, etc. Conversion of these may or may not work depending on just how they are represented internally in the PDF.
- Links and Tables of Contents are not supported
- PDFs that use embedded non-unicode fonts to represent non-English characters will result in garbled output for those characters
- Some PDFs are made up of photographs of the page with OCRed text behind them. In such cases calibre uses the OCRed text, which can be very different from what you see when you view the PDF file
- PDFs that are used to display complex text, like right to left languages and math typesetting will not convert correctly

To re-iterate **PDF is a really, really bad** format to use as input. If you absolutely must use PDF, then be prepared for an output ranging anywhere from decent to unusable, depending on the input PDF.

Comic Book Collections

A comic book collection is a .cbc file. A .cbc file is a zip file that contains other CBZ/CBR files. In addition the .cbc file must contain a simple text file called comics.txt, encoded in UTF-8. The comics.txt file must contain a list of the comics files inside the .cbc file, in the form filename:title, as shown below:

```
one.cbz:Chapter One
two.cbz:Chapter Two
three.cbz:Chapter Three
```

The .cbc file will then contain:

```
comics.txt
one.cbz
two.cbz
three.cbz
```

calibre will automatically convert this .cbc file into a ebook with a Table of Contents pointing to each entry in comics.txt.

EPUB advanced formatting demo

Various advanced formatting for EPUB files is demonstrated in this demo file[26]. The file was created from hand coded HTML using calibre and is meant to be used as a template for your own EPUB creation efforts.

The source HTML it was created from is available demo.zip[27]. The settings used to create the EPUB from the ZIP file are:

```
ebook-convert demo.zip .epub -vv --authors "Kovid Goyal" --language en --level1-toc '//*[@class="titl
```

Note that because this file explores the potential of EPUB, most of the advanced formatting is not going to work on readers less capable than calibre's built-in EPUB viewer.

Convert ODT documents

calibre can directly convert ODT (OpenDocument Text) files. You should use styles to format your document and minimize the use of direct formatting. When inserting images into your document you need to anchor them to the paragraph, images anchored to a page will all end up in the front of the conversion.

To enable automatic detection of chapters, you need to mark them with the build-in styles called 'Heading 1', 'Heading 2', ..., 'Heading 6' ('Heading 1' equates to the HTML tag <h1>, 'Heading 2' to <h2> etc). When you convert in calibre you can enter which style you used into the 'Detect chapters at' box. Example:

- If you mark Chapters with style 'Heading 2', you have to set the 'Detect chapters at' box to `//h:h2`
- For a nested TOC with Sections marked with 'Heading 2' and the Chapters marked with 'Heading 3' you need to enter `//h:h2|//h:h3`. On the Convert - TOC page set the 'Level 1 TOC' box to `//h:h2` and the 'Level 2 TOC' box to `//h:h3`.

Well-known document properties (Title, Keywords, Description, Creator) are recognized and calibre will use the first image (not to small, and with good aspect-ratio) as the cover image.

There is also an advanced property conversion mode, which is activated by setting the custom property `opf.metadata` ('Yes or No' type) to Yes in your ODT document (File->Properties->Custom Properties). If this property is detected by calibre, the following custom properties are recognized (`opf.authors` overrides document creator):

```
opf.titlesort
opf.authors
opf.authorsort
opf.publisher
opf.pubdate
opf.isbn
opf.language
opf.series
opf.seriesindex
```

In addition to this, you can specify the picture to use as the cover by naming it `opf.cover` (right click, Picture->Options->Name) in the ODT. If no picture with this name is found, the 'smart' method is used. As the cover detection might result in double covers in certain output formats, the process will remove the paragraph (only if the only content is the cover!) from the document. But this works only with the named picture!

To disable cover detection you can set the custom property `opf.nocover` ('Yes or No' type) to Yes in advanced mode.

[26] http://calibre-ebook.com/downloads/demos/demo.epub
[27] http://calibre-ebook.com/downloads/demos/demo.zip

Converting to PDF

The first, most important, setting to decide on when converting to PDF is the page size. By default, calibre uses a page size defined by the current *Output profile*. So if your output profile is set to Kindle, calibre will create a PDF with page size suitable for viewing on the small kindle screen. However, if you view this PDF file on a computer screen, then it will appear to have too large fonts. To create "normal" sized PDFs, use the *Override page size* option under *PDF Output* in the conversion dialog.

Headers and Footers

You can insert arbitrary headers and footers on each page of the PDF by specifying header and footer templates. Templates are just snippets of HTML code that get rendered in the header and footer locations. For example, to display page numbers centered at the bottom of every page, in green, use the following footer template:

```
<p style="text-align:center; color:green">Page _PAGENUM_</p>
```

calibre will automatically replace _PAGENUM_ with the current page number. You can even put different content on even and odd pages, for example the following header template will show the title on odd pages and the author on even pages:

```
<p style="text-align:right"><span class="even_page">_AUTHOR_</span><span class="odd_page"><i>_TITLE_<
```

calibre will automatically replace _TITLE_ and _AUTHOR_ with the title and author of the document being converted. You can also display text at the left and right edges and change the font size, as demonstrated with this header template:

```
<div style="font-size:x-small"><p style="float:left">_TITLE_</p><p style="float:right;"><i>_AUTHOR_</
```

This will display the title at the left and the author at the right, in a font size smaller than the main text.

You can also use the current section in templates, as shown below:

```
<p style="text-align:right">_SECTION_</p>
```

SECTION is replaced by whatever the name of the current section is. These names are taken from the metadata Table of Contents in the document (the PDF Outline). If the document has no table of contents then it will be replaced by empty text. If a single PDF page has multiple sections, the first section on the page will be used.

You can even use javascript inside the header and footer templates, for example, the following template will cause page numbers to start at 4 instead of 1:

```
<p id="pagenum" style="text-align:center;"></p><script>document.getElementById("pagenum").innerHTML =
```

Poznámka: When adding headers and footers make sure you set the page top and bottom margins to large enough values, under the Page Setup section of the conversion dialog.

Printable Table of Contents

You can also insert a printable Table of Contents at the end of the PDF that lists the page numbers for every section. This is very useful if you intend to print out the PDF to paper. If you wish to use the PDF on an electronic device, then the PDF Outline provides this functionality and is generated by default.

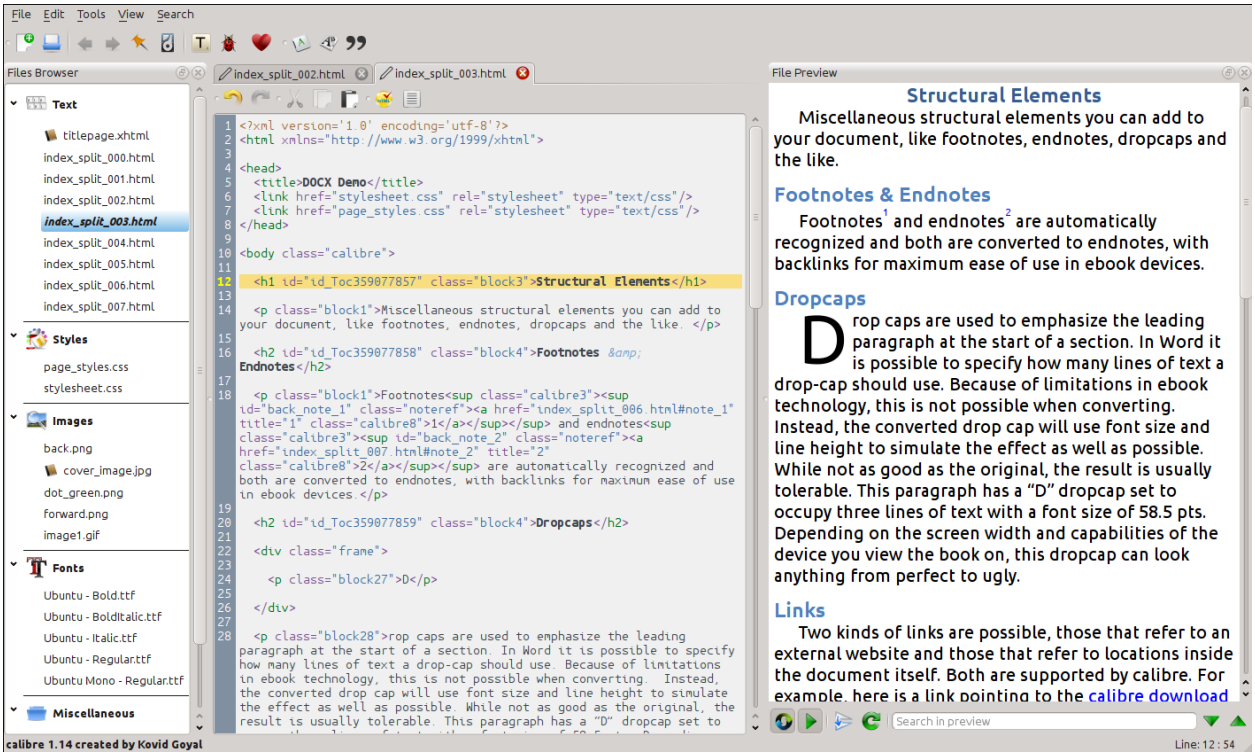
You can customize the look of the the generated Table of contents by using the Extra CSS conversion setting under the Look & Feel part of the conversion dialog. The default css used is listed below, simply copy it and make whatever changes you like.

```
.calibre-pdf-toc table { width: 100%% }

.calibre-pdf-toc table tr td:last-of-type { text-align: right }

.calibre-pdf-toc .level-0 {
    font-size: larger;
}

.calibre-pdf-toc .level-1 td:first-of-type { padding-left: 1.4em }
.calibre-pdf-toc .level-2 td:first-of-type { padding-left: 2.8em }
```

1.5 Úprava e-knih

Calibre má integrovaný editor e-knih, který může být použit pro úpravu knih ve formátech EPUB a AZW3 (Kindle). Editor vám zobrazí HTML a CSS, které jsou použity uvnitř souborů knih s živým náhledem, který se aktualizace, zatímco provádíte změny. Také obsahuje různé automatizované nástroje pro provedení úkolů běžného vyčištění a oprav.

Tento editor můžete použít kliknutím pravým tlačítkem myši na kteroukoliv knihu v Calibre a výběrem *Upravit knihu*.


File Edit Tools View Search
Files Browser
Text
titlepage.xhtml
index_split_000.html
index_split_001.html
index_split_002.html
index_split_003.html
index_split_004.html
index_split_005.html
index_split_006.html
index_split_007.html
Styles
page_styles.css
stylesheet.css
Images
back.png
cover_image.jpg
dot_green.png
forward.png
image1.gif
Fonts
Ubuntu - Bold.ttf
Ubuntu - BoldItalic.ttf
Ubuntu - Italic.ttf
Ubuntu - Regular.ttf
Ubuntu Mono - Regular.ttf
Miscellaneous
calibre 1.14 created by Kovid Goyal
index_split_002.html
index_split_003.html
File Preview
Structural Elements
Miscellaneous structural elements you can add to your document, like footnotes, endnotes, dropcaps and the like.
Footnotes & Endnotes
Footnotes[1] and endnotes[2] are automatically recognized and both are converted to endnotes, with backlinks for maximum ease of use in ebook devices.
Dropcaps
Drop caps are used to emphasize the leading paragraph at the start of a section. In Word it is possible to specify how many lines of text a drop-cap should use. Because of limitations in ebook technology, this is not possible when converting. Instead, the converted drop cap will use font size and line height to simulate the effect as well as possible. While not as good as the original, the result is usually tolerable. This paragraph has a "D" dropcap set to occupy three lines of text with a font size of 58.5 pts. Depending on the screen width and capabilities of the device you view the book on, this dropcap can look anything from perfect to ugly.
Links
Two kinds of links are possible, those that refer to an external website and those that refer to locations inside the document itself. Both are supported by calibre. For example, here is a link pointing to the calibre download
Search in preview
Line: 12 : 54

Obsah

- *Základní pracovní postup* (stránka 72)
- *Prohlížeč souborů* (stránka 75)
  - *Přejmenovávání souborů* (stránka 76)
  - *Slučování souborů* (stránka 76)
  - *Změna pořadí textových souborů* (stránka 76)
  - *Označování obálky* (stránka 76)
  - *Odstraňování souborů* (stránka 77)
  - *Exportování souborů* (stránka 77)
  - *Přidávání nových obrázků, písem atd. nebo vytváření nových prázdných souborů* (stránka 77)
  - *Nahrazování souborů* (stránka 77)
  - *Efektivní propojování šablon stylů se soubory HTML* (stránka 77)
- *Hledat a nahradit* (stránka 77)
  - *Uložená hledání* (stránka 78)
  - *Režim funkcí* (stránka 78)
- *Automatizované nástroje* (stránka 78)
  - *Upravit obsah* (stránka 78)
  - *Zkontrolovat knihu* (stránka 80)
  - *Přidat obálku* (stránka 81)
  - *Vkládání odkazovaných písem* (stránka 81)
  - *Částečné vkládání písem* (stránka 81)
  - *Vylepšení interpunkce* (stránka 81)
  - *Odebírání nepoužívaných pravidel CSS* (stránka 81)
  - *Opravit HTML* (stránka 82)
  - *Zkrášlování souborů* (stránka 82)
  - *Vložit vložený obsah* (stránka 82)
  - *Nastavit sémantiku* (stránka 82)
  - *Filtrovat informace o stylu* (stránka 82)
- *Kontrolní body* (stránka 83)
- *Panel Živý náhled* (stránka 85)
  - *Rozdělování souborů HTML* (stránka 86)
- *Panel Živé CSS* (stránka 87)
- *Různé nástroje* (stránka 88)
  - *Zobrazení obsahu* (stránka 88)
  - *Kontrola pravopisu slov v knize* (stránka 88)
  - *Vkládání speciálních znaků* (stránka 90)
  - *Zobrazení kontroly kódu* (stránka 90)
  - *Kontrola externích odkazů* (stránka 90)
  - *Uspořádání souborů do složek podle typu* (stránka 91)
  - *Importování souborů v jiných formátech e-knih než EPUB* (stránka 91)
  - *Nástroj Hlášení* (stránka 99)
- *Speciální funkce v editoru kódu* (stránka 99)
  - *Zvýraznění syntaxe* (stránka 99)
  - *Nápověda závislá na kontextu* (stránka 99)
  - *Automatické dokončování* (stránka 100)
  - *Fragmenty* (stránka 100)

1.5.1 Základní pracovní postup

Poznámka: Videoprohlídka editoru Calibre je dostupná zde[28].

[28] http://calibre-ebook.com/demo#tutorials

Když poprvé otevřete nástrojem pro úpravu knihy, bude vám vlevo nabídnut seznam souborů. Jsou to jednotlivé soubory HTML, šablony stylů, obrázky atd., které tvoří obsah knihy. Jednoduše poklepejte na soubor pro spuštění jeho úpravy. Pamatujte, že pokud chcete udělat něco složitějšího než provedení několika malých vylepšení, budete potřebovat znát Kurz HTML[29] a Kurz CSS[30].

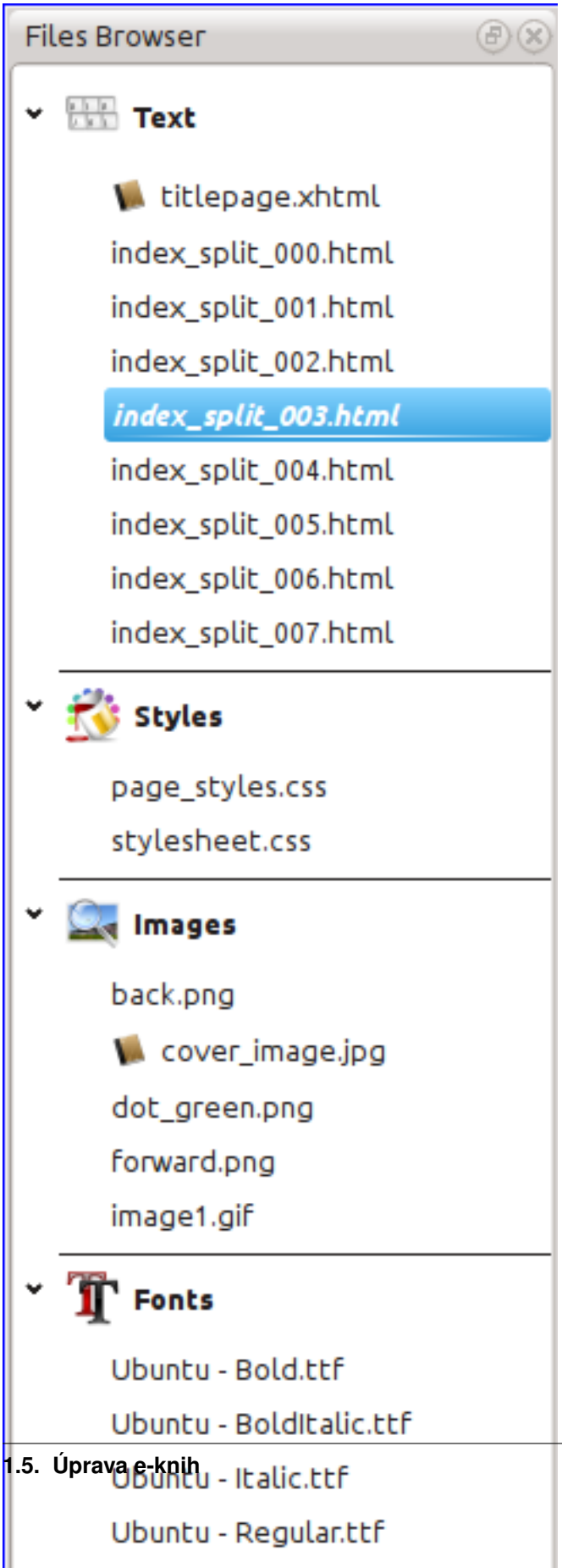
Zatímco provádíte v editoru změny v HTML nebo CSS, bude se vpravo v panelu náhledu zobrazovat živý náhled změn. Pokud jste spokojeni s tím, jak vypadají změny, které jste provedli, klikněte na tlačítko Uložit nebo použijte *Soubor->Uložit* pro uložení změn do e-knihy.

Jednou z užitečných funkcí jsou *Kontrolní body*. Než se pustíte do nějaké ambiciózní sady úprav, můžete vytvořit kontrolní bod. Kontrolní bod zachová aktuální stav vaší knihy, pak pokud se v budoucnu rozhodnete, že se vám nelíbí změny, které jste provedli, můžete se vrátit do stavu, kdy jste vytvořili kontrolní bod. Abyste vytvořili kontrolní bod, použijte *Úpravy->Vytvořit kontrolní bod*. Kontrolní body budou pro vás také automaticky vytvořeny, kdykoliv spustíte jakýkoliv automatizovaný nástroj, jako je globální hledání a nahrazování. Funkce kontrolních bodů je doplněk k normálnímu mechanismu Zpět/Znovu při úpravách jednotlivých souborů. Kontrolní body jsou užitečné, když jsou změny rozděleny do více souborů v knize.

Toto je základní pracovní postup pro úpravy knih – Otevřete soubor, proveďte změny, zkontrolujte je a uložte. Zbytek této příručky probere různé přítomné nástroje a funkce, které vám umožní efektivně provést konkrétní úkoly.

[29] http://html.net/tutorials/html/
[30] http://html.net/tutorials/css/

1.5.2 Prohlížeč souborů



Prohlížeč souborů vám poskytuje přehled o různých souborech uvnitř knihy, kterou upravujete. Soubory jsou uspořádány podle kategorie s textovými soubory (HTML) nahoře, následovanými soubory šablon stylů (CSS), obrázky a tak dále. Jednoduše poklepejte na soubor pro spuštění jeho úpravy. Úprava je podporována pro soubory HTML, CSS a obrázky. Pořadí textových souborů je stejné, v jakém by byly zobrazeny, pokud byste četli knihu. Všechny ostatní soubory jsou uspořádány abecedně.

Přejetím myší nad položkou můžete vidět její velikost a také úplnou cestu k souboru uvnitř knihy ve spodní části obrazovky. Pamatujte, že soubory uvnitř e-knih jsou komprimované, takže velikost konečné knihy není součtem jednotlivých velikostí souborů.

Mnoho souborů má v knize zvláštní význam. Ty budou mít obvykle vedle svých názvů ikonu označující zvláštní význam. Například na obrázku vlevo můžete vidět, že soubory *cover_image.jpg* a *titlepage.xhtml* mají vedle sebe ikonu obálky, to znamená, že jsou to obrázek obálky knihy a titulní stránka. Podobně soubor *content.opf* má vedle sebe ikonu metadat, která označuje, že obsahuje metadata knihy, a soubor *toc.ncx* má vedle sebe ikonu T označující, že je to Obsah.

Na jednotlivých souborech můžete provádět mnoho akcí kliknutím pravým tlačítkem myši na ně.

Přejmenovávání souborů

Jednotlivý soubor můžete přejmenovat kliknutím pravým tlačítkem myši na něj a výběrem *Přejmenovat*. Přejmenování souboru automaticky aktualizuje všechny odkazy na něj v celé knize. Takže musíte pouze zadat nový název, Calibre se postará o zbytek.

Můžete také hromadně přejmenovat mnoho souborů najednou. To je užitečné, pokud chcete, aby soubory měly nějaký jednoduchý vzor názvu. Například můžete chtít přejmenovat všechny soubory HTML, aby měly názvy Kapitola-1.html, Kapitola-2.html a tak dále. Vyberte soubory, které chcete hromadně přejmenovat, podržením klávesy Shift nebo Ctrl a kliknutím na soubory. Pak klikněte pravým tlačítkem myši a vyberte *Hromadně přejmenovat vybrané soubory*. Zadejte předponu a číslo, kterým má začínat automatické číslování, klikněte na OK a je hotovo.

Slučování souborů

Někdy můžete chtít sloučit dohromady dva soubory HTML nebo dva soubory CSS. Někdy může být užitečné mít vše v jednom souboru. Ale buďte opatrní, uložení velkého množství obsahu do jednoho souboru způsobí problémy s výkonem při zobrazování knihy v typické čtečce e-knih.

Pro sloučení více souborů dohromady je vyberte podržením klávesy Ctrl a kliknutím na ně (ujistěte se, že vybíráte pouze soubory jednoho typu, buď všechny soubory HTML nebo všechny soubory CSS a tak dále). Pak klikněte pravým tlačítkem myši a vyberte Sloučit. To je všechno, Calibre sloučí soubory, automaticky se postará o migraci všech odkazů na sloučené soubory. Pamatujte, že slučování souborů může někdy způsobit změnu stylu textů, protože jednotlivé soubory mohly používat různé šablony stylů.

Změna pořadí textových souborů

Pořadí, ve kterém jsou textové soubory (HTML) otevírány při čtení knihy, můžete přeuspořádat jednoduše jejich přetažením do Prohlížeče souborů. Pro technicky nakloněné, tomu se říká znovu uspořádání páteře knihy. Pamatujte, že musíte přetáhnout položky *mezi* jiné položky, ne nad ně, to může být trochu nešikovné, dokud si na to nezvyknete.

Označování obálky

E-knihy mají obvykle obrázek obálky. Tento obrázek je označen v Prohlížeči souborů ikonou hnědé knihy vedle názvu obrázku. Pokud chcete určit nějaký jiný obrázek jako obálku, můžete to provést kliknutím pravým tlačítkem myši na soubor a výběrem *Označit jako obrázek obálky*.

Navíc soubory EPUB mají koncept *titulní stránky*. Titulní stránka je soubor HTML, který se chová jako titulní stránka nebo obálka knihy. Označit soubor HTML jako titulní stránku při úpravě souborů EPUB můžete kliknutím pravým tlačítkem myši. Dávejte pozor na to, aby označený soubor obsahoval pouze informace o obálce. Pokud obsahuje jiný obsah, jako je první kapitola, pak bude tento obsah ztracen, pokud uživatel někdy převede soubor EPUB v Calibre do jiného formátu. To je proto, že při převodu Calibre předpokládá, že označená titulní stránka obsahuje pouze obálku a žádný jiný obsah.

Odstraňování souborů

Soubory můžete odstranit buď kliknutím pravým tlačítkem myši na ně, nebo jejich výběrem a stisknutím klávesy Delete. Odstranění souboru odebere všechny odkazy na soubor ze souboru OPF, což vám ušetří tu práci. Odkazy na jiných místech však odebrány nejsou, pro jejich snadné nalezení a odebrání nebo nahrazení můžete použít nástroj Zkontrolovat knihu.

Exportování souborů

Soubor zevnitř knihy můžete exportovat někam jinam ve vašem počítači. To je užitečné, pokud chcete pracovat na souboru izolovaně pomocí specializovaných nástrojů. Abyste to udělali, jednoduše klikněte pravým tlačítkem myši na soubor a zvolte *Exportovat*.

Jakmile dokončíte práci na exportovaném souboru, můžete ho znovu importovat do knihy novým kliknutím pravým tlačítkem myši na soubor a výběrem *Nahradit souborem...*, což vám umožní nahradit soubor v knize dříve exportovaným souborem.

Přidávání nových obrázků, písem atd. nebo vytváření nových prázdných souborů

Přidat nový obrázek, písmo, šablonu stylů atd. z vašeho počítače do knihy můžete kliknutím na *Soubor->Nový soubor*. To vám umožní buď importovat soubor kliknutím na tlačítko *Importovat zdrojový soubor*, nebo vytvořit nový prázdný soubor HTML nebo šablonu stylů jednoduše zadáním názvu souboru do pole pro nový soubor.

Můžete také importovat více souborů do knihy najednou pomocí Soubor-> Importovat soubory do knihy.

Nahrazování souborů

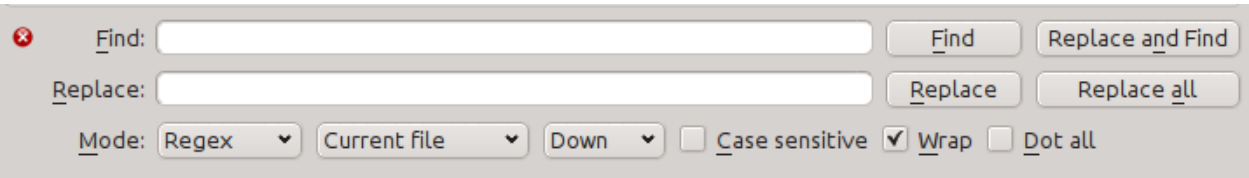
Existující soubory v knize můžete snadno nahradit kliknutím pravým tlačítkem myši na soubor a výběrem nahradit. Tím se automaticky aktualizují všechny odkazy a reference, pokud má nahrazující soubor jiný název než nahrazený soubor.

Efektivní propojování šablon stylů se soubory HTML

Pro větší pohodlí můžete v Prohlížeči souborů vybrat více souborů HTML, kliknout pravým tlačítkem myši a vybrat Propojit šablony stylů, aby Calibre automaticky vložilo značky <link> pro tyto šablony stylů do všech vybraných souborů HTML.

1.5.3 Hledat a nahradit

Upravit knihu má velice výkonné rozhraní hledání a nahrazení, které vám umožňuje hledat a nahradit text v aktuálním souboru, napříč všemi soubory a dokonce i v označené oblasti aktuálního souboru. Hledat můžete pomocí normálního hledání nebo pomocí regulárních výrazů. Abyste zjistili, jak používat regulární výrazy pro pokročilé vyhledávání, podívejte se na *Vše o používání regulárních výrazů v Calibre* (stránka 163).



Hledání a nahrazení spusťte pomocí položky nabídky *Hledat->Najít a nahradit* (musíte upravovat soubor HTML nebo CSS).

Napište text, který chcete najít, do pole Najít a jeho nahrazení do pole Nahradit. Můžete kliknout na příslušná tlačítka pro nalezení další shody, nahrazení aktuální shody a nahrazení všech shod.

Pomocí rozevíracích nabídek v dolní části okna můžete nechat hledání pracovat nad aktuálním souborem, všemi textovými soubory, všemi soubory stylů nebo všemi soubory. Můžete také zvolit režim hledání, aby byl normální (řetězec) nebo regulární výraz.

Všechny shody pro hledaný výraz můžete spočítat pomocí *Hledat->Spočítat vše*. Počítání se spustí nad všemi soubory nebo oblastmi, které jste vybrali v rozevíracím poli.

Můžete také přejít na konkrétní řádek v aktuálně otevřeném editoru pomocí *Hledat->Přejít na řádek*.

Poznámka: Nezapomeňte, že pro využití plného výkonu hledání a nahrazení budete muset použít regulární výrazy. Podívejte se na *Vše o používání regulárních výrazů v Calibre* (stránka 163).

Uložená hledání

Často používané výrazy hledání nebo nahrazení můžete uložit a vícekrát je znovu použít . Pro uložení hledání jednoduše klikněte pravým tlačítkem myši do pole Najít a vyberte *Uložit aktuální hledání*.

Dialogové okno uložených hledání můžete vyvolat pomocí *Hledat->Uložená hledání*. To zobrazí seznam výrazů hledání a nahrazení, které můžete použít. Můžete dokonce vybrat více položek v seznamu podržením klávesy Ctrl při kliknutí, aby se spustilo více výrazů hledání a nahrazení v jedné operaci.

Režim funkcí

Režim funkcí vám umožňuje psát libovolně výkonné funkce Pythonu, které jsou spuštěny na každém Najít a nahradit. V režimu funkcí můžete provést skoro jakékoliv zpracování textu, jaké chcete. Další informace naleznete v Function Mode for Search & Replace in the Editor (stránka 91).

1.5.4 Automatizované nástroje

Editor knih má různé nástroje pro pomoc s běžnými úkoly. Ty jsou přístupné pomocí nabídky *Nástroje*.

Upravit obsah

Pro usnadnění úpravy obsahu je zde specializovaný nástroj. Spustíte ho pomocí *Nástroje->Obsah->Upravit obsah*.

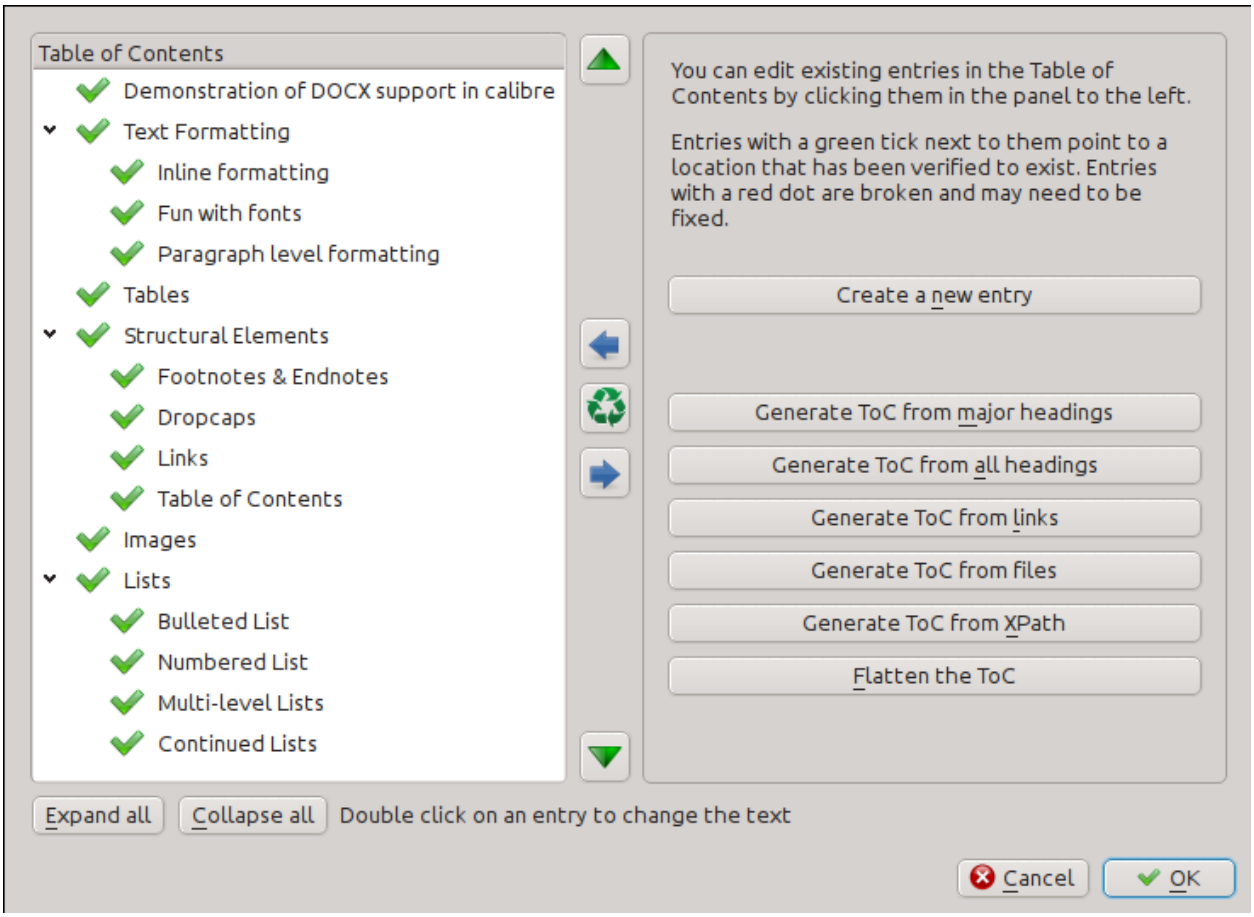
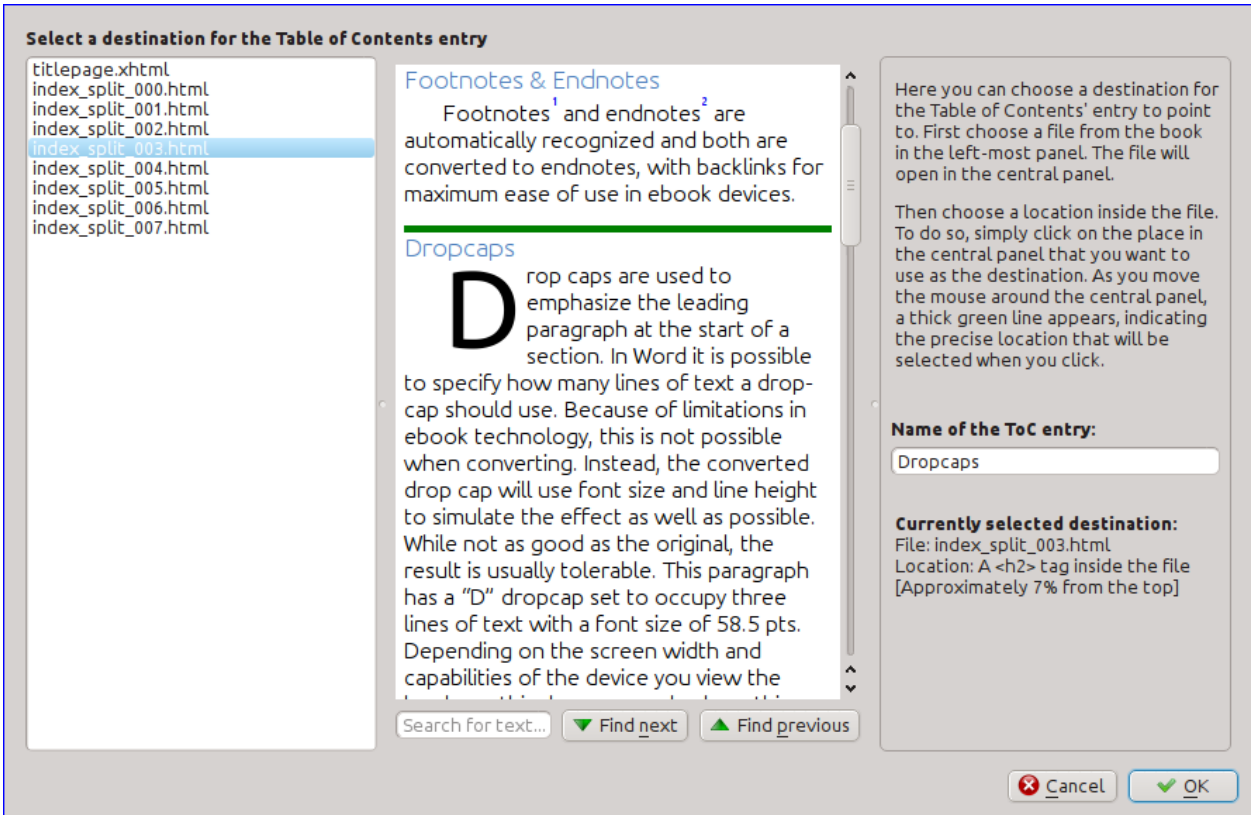


Table of Contents
Demonstration of DOCX support in calibre
Text Formatting
Inline formatting
Fun with fonts
Paragraph level formatting
Tables
Structural Elements
Footnotes & Endnotes
Dropcaps
Links
Table of Contents
Images
Lists
Bulleted List
Numbered List
Multi-level Lists
Continued Lists
You can edit existing entries in the Table of Contents by clicking them in the panel to the left.
Entries with a green tick next to them point to a location that has been verified to exist. Entries with a red dot are broken and may need to be fixed.
Create a new entry
Generate ToC from major headings
Generate ToC from all headings
Generate ToC from links
Generate ToC from files
Generate ToC from XPath
Flatten the ToC
Expand all
Collapse all
Double click on an entry to change the text
Cancel
OK

Nástroj Upravit obsah zobrazí aktuální Obsah (pokud existuje) vlevo. Jednoduše poklepejte na jakoukoliv položku pro změnu jejího textu. Můžete také přeuspořádat položky přetažením nebo pomocí tlačítek vpravo.

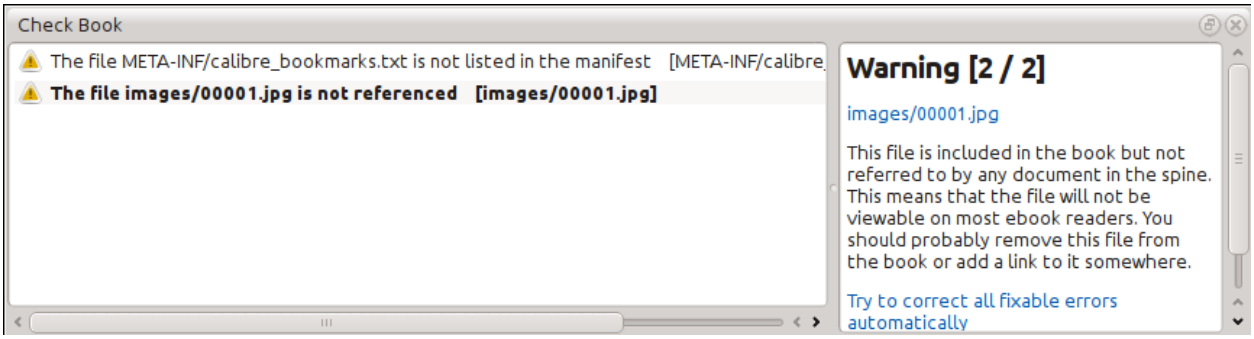
U knih, které nemají již existující Obsah, vám nástroj dává různé možnosti automatického vygenerování obsahu z textu. Můžete ho vygenerovat z nadpisů v dokumentu, z odkazů, z jednotlivých souborů a tak dále.

Jednotlivé položky můžete upravit kliknutím na ně a pak kliknutím na tlačítko *Změnit umístění, na které tato položka ukazuje*. To otevře malý náhled knihy, jednoduše přesuňte kurzor myši nad panel zobrazení knihy a klikněte na umístění, kam chcete, aby položka ukazovala. Tlustá zelená čára vám ukáže umístění. Klikněte na OK, jakmile jste spokojeni s umístěním.


Select a destination for the Table of Contents entry
titlepage.xhtml
index_split_000.html
index_split_001.html
index_split_002.html
index_split_003.html
index_split_004.html
index_split_005.html
index_split_006.html
index_split_007.html
Footnotes & Endnotes
Footnotes[1] and endnotes[2] are automatically recognized and both are converted to endnotes, with backlinks for maximum ease of use in ebook devices.
Dropcaps
Drop caps are used to emphasize the leading paragraph at the start of a section. In Word it is possible to specify how many lines of text a drop-cap should use. Because of limitations in ebook technology, this is not possible when converting. Instead, the converted drop cap will use font size and line height to simulate the effect as well as possible. While not as good as the original, the result is usually tolerable. This paragraph has a "D" dropcap set to occupy three lines of text with a font size of 58.5 pts. Depending on the screen width and capabilities of the device you view the
Search for text...
Find next
Find previous
Here you can choose a destination for the Table of Contents' entry to point to. First choose a file from the book in the left-most panel. The file will open in the central panel.
Then choose a location inside the file. To do so, simply click on the place in the central panel that you want to use as the destination. As you move the mouse around the central panel, a thick green line appears, indicating the precise location that will be selected when you click.
Name of the ToC entry:
Dropcaps
Currently selected destination:
File: index_split_003.html
Location: A <h2> tag inside the file
[Approximately 7% from the top]
Cancel
OK

Zkontrolovat knihu

Nástroj *Zkontrolovat knihu* vyhledá ve vaší knize problémy, které by mohly bránit zamýšlenému fungování na skutečných čtecích zařízeních. Aktivujte ho pomocí *Nástroje->Zkontrolovat knihu*.


Check Book
The file META-INF/calibre_bookmarks.txt is not listed in the manifest [META-INF/calibre
The file images/00001.jpg is not referenced [images/00001.jpg]
Warning [2 / 2]
images/00001.jpg
This file is included in the book but not referred to by any document in the spine. This means that the file will not be viewable on most ebook readers. You should probably remove this file from the book or add a link to it somewhere.
Try to correct all fixable errors automatically

Všechny nalezené problémy jsou nahlášeny v pěkném, snadno použitelném seznamu. Kliknutí na jakoukoliv položku v seznamu zobrazí nějakou nápovědu k této chybě a také vám dá možnost tuto chybu automaticky opravit, pokud chyba může být automaticky opravena. Můžete také poklepat na chybu pro otevření umístění chyby v editoru, abyste ji mohli opravit sami.

Některé z prováděných kontrol jsou:

- Poškozené značení HTML. Je nahlášeno jakékoliv značení HTML, které není analyzována jako dobře vytvořené XML. Oprava zajistí, že vaše značení funguje ve všech kontextech tak, jak má. Calibre také může automaticky opravit tyto chyby, ale automatická oprava může mít někdy neočekávané účinky, proto používejte opatrně. Jako vždy je před automatickou opravou vytvořen kontrolní bod, takže můžete snadno vrátit všechny změny. Automatická oprava funguje na základě analýzy značení pomocí algoritmu HTML5, který je vysoce odolný proti

chybám, a pak převodem na dobře vytvořené XML.

- Poškozené nebo neznámé styly CSS. Je nahlášeno jakékoliv CSS, které není platné, nebo které má vlastnosti nedefinované ve standardu CSS 2.1 (a několik z CSS 3). CSS je kontrolováno ve všech šablonách stylů, vložených atributů stylů a značkách <style> v souborech HTML.
- Nefunkční odkazy. Jsou nahlášeny odkazy, které ukazují na soubory uvnitř knihy, které chybí.
- Neodkazované soubory. Jsou nahlášeny soubory v knize, na které neodkazuje žádný jiný soubor, nebo nejsou v páteři.
- Různé běžné problémy v souborech OPF, jako jsou duplicitní položky páteře nebo manifestu, poškozené atributy idref nebo metaznačky obálek, chybějící požadované oddíly a tak dále.
- Různé kontroly kompatibility na známé problémy, které mohou způsobit nefunkčnost knihy ve čtecích zařízeních.

Přidat obálku

Přidat obálku knize můžete snadno pomocí *Nástroje->Přidat obálku*. To vám umožní vybrat si jako obálku existující obrázek v knize nebo importovat nový obrázek do knihy a udělat z něj obálku. Při úpravě souborů EPUB je pro obálku automaticky vygenerováno obalující HTML. Pokud je v knize nalezena existující obálka, je nahrazena. Nástroj se také automaticky postará o správné označení souborů obálek jako obálek v OPF.

Vkládání odkazovaných písem

Přístupné pomocí *Nástroje->Vložit odkazovaná písma*. Tento nástroj najde všechna písma odkazovaná v knize, a pokud již nejsou vložena, vyhledá je ve vašem počítači a vloží je do knihy, pokud jsou nalezena. Ujistěte se, že máte potřebná autorská práva pro vkládání komerčně licencovaných písem, než to uděláte.

Částečné vkládání písem

Přístupné pomocí *Nástroje->Částečně vložit písma*. Tento nástroj zmenší všechna písma v knize tak, aby obsahovala pouze glyfy pro text skutečně přítomný v knize. Toto běžně zmenší velikost souborů písem o asi 50 %. Je však třeba si uvědomit, že jakmile jsou písma vložena částečně, tak pokud přidáte nový text, jehož znaky nejsou přítomné v částečně vloženém písmu, písmo nebude pro nový text fungovat. Takže toto dělejte pouze jako poslední krok ve vašem pracovním postupu.

Vylepšení interpunkce

Převede pomlčky, tři tečky, uvozovky, vícenásobné spojovníky atd. v prostém textu na jejich typograficky správné ekvivalenty. Pamatujte, že algoritmus může někdy generovat nesprávné výsledky, obzvláště pokud jde o jednoduché uvozovky na začátku zkrácených tvarů. Přístupné pomocí *Nástroje->Vylepšit interpunkci*.

Odebírání nepoužívaných pravidel CSS

Odebere všechna nepoužívaná pravidla CSS ze šablon stylů a značek <style>. Některé knihy vytvořené pomocí výrobních šablon mohou mít velký počet nadbytečných pravidel CSS, která neodpovídají žádnému skutečnému obsahu. Tato nadbytečná pravidla mohou zpomalit čtečky, které je musejí všechny zpracovat. Přístupné pomocí *Nástroje->Odebrat nepoužívaná pravidla CSS*.

Opravit HTML

Tento nástroj jednoduše převádí HTML, které nemůže být analyzováno jako XML, na dobře vytvořené XML. U e-knih je velice běžné, že nemají dobře vytvořené XML, takže tento nástroj jednoduše automatizuje proces opravy takového HTML. Nástroj funguje analýzou HTML pomocí algoritmu HTML5 (algoritmus použitý ve všech moderních prohlížečích) a poté převedením výsledek na XML. Uvědomte si, že automatické opravy mohou někdy mít neintuitivní výsledky. Pokud tomu dáváte přednost, můžete použít výše probíraný nástroj Zkontrolovat knihu k nalezení a ruční opravě problémů v HTML. Přístupné pomocí *Nástroje->Opravit HTML*.

Zkrášlování souborů

Tento nástroj se používá pro automatické formátování všech souborů HTML a CSS tak, aby "vypadali pěkně". Kód je automaticky odsazený, takže je pěkně řazený, v případě potřeby jsou vloženy prázdné řádky a tak dále. Pamatujte, že zkrášlování také automaticky opraví poškozené HTML nebo CSS. Proto pokud nechcete, aby byly provedeny jakékoliv automatické opravy, použijte nejdříve nástroj Zkontrolovat knihu pro opravu všech problémů, a teprve pak spusťte zkrášlení. Přístupné pomocí *Nástroje->Zkrášlit všechny soubory*.

Poznámka: V HTML může mít jakýkoliv text významové prázdné znaky podle směrnice CSS o prázdných znacích. Proto by mohlo zkrášlení změnit vykreslení HTML. Aby se tomu zabránilo, pokud je to možné, algoritmus zkrášlení zkrášluje pouze značky na úrovni bloků, které obsahují jiné značky na úrovni bloků. Takže například text uvnitř značky <p> nebude mít změněny prázdné znaky. Ale značka <body>, která obsahuje pouze další značky <p> a , bude zkrášlena. To může někdy znamenat, že určitý soubor nebude zkrášlením ovlivněn, protože nemá žádné vhodné značky na úrovni bloků. V takových případech můžete zkusit různé nástroje zkrášlení, které jsou méně opatrné, například: *Úklid HTML Tidy <http://infohound.net/tidy/> _*.

Vložit vložený obsah

Normálně je v e-knihách obsah oddělen od hlavního textu a je obvykle přístupný prostřednictvím speciálního tlačítka nebo nabídky Obsah ve čtecím zařízení e-knih. Můžete také nechat Calibre automaticky vygenerovat *vložený* obsah, který se stane součástí textu knihy. Je generovaný na základě aktuálně definovaného obsahu.

Pokud tento nástroj použijete vícekrát, každé vyvolání způsobí že dříve vytvořený vložený obsah bude nahrazen. Nástroj je přístupný pomocí *Nástroje->Obsah->Vložit vložený obsah*.

Nastavit sémantiku

Tento nástroj se používá k nastavení *sémantiky* v souborech EPUB. Sémantika jsou jednoduše odkazy v souboru OPF, které identifikují určitá umístění v knize, že mají zvláštní význam. Můžete je použít k identifikaci předmluvy, věnování, obálky, obsahu atd. Jednoduše zvolte typ sémantické informace, který chcete zadat, a pak vyberte umístění v knize, na které má odkaz směřovat. Tento nástroj je přístupný pomocí *Nástroje->Nastavit sémantiku*.

Filtrovat informace o stylu

Tento nástroj může být použit pro snadné odebrání zadaných vlastností stylů CSS z celé knihy. Můžete mu říct, které vlastnosti chcete odebrat, například `color, background-color, line-height`, a budou odebrány ze všech míst, kde se vyskytují — šablony stylů, značky `<style>` a vložené atributy `style`. Po odebrání informací o stylu se zobrazí souhrn všech provedených změn, takže můžete vidět, co přesně se změnilo. Nástroj je přístupný pomocí *Nástroje->Filtrovat informace o stylu*.

1.5.5 Kontrolní body

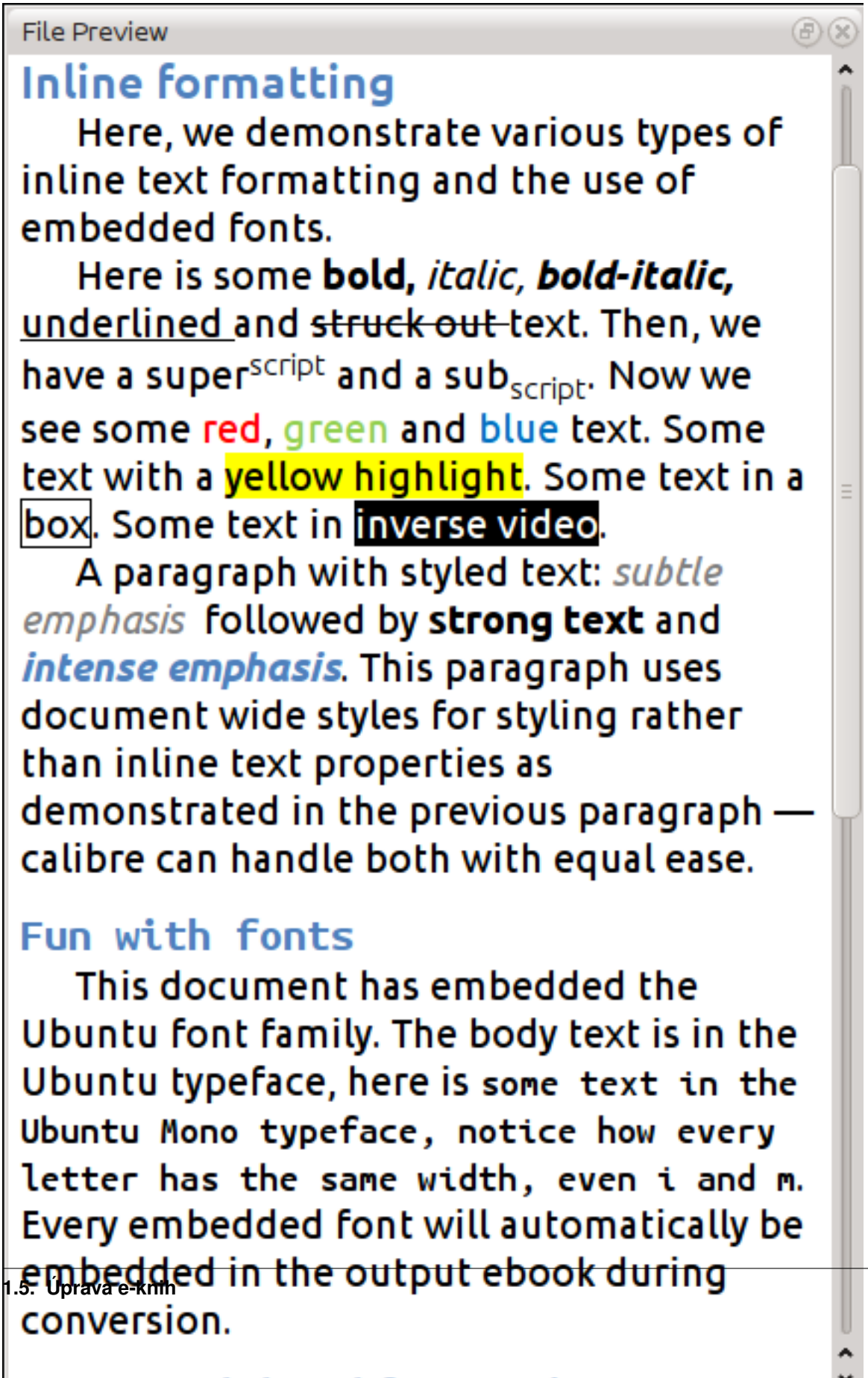
Kontrolní body jsou způsob, jak označit aktuální stav knihy jako "speciální". Pak můžete pokračovat v provádění jakýchkoliv změn v knize, a pokud se vám nelíbí výsledky, můžete se vrátit do stavu kontrolního bodu. Kontrolní body jsou automaticky vytvořeny při každém spuštění některého z automatizovaných nástrojů popsaných v předchozí části.

Kontrolní bod můžete vytvořit pomocí *Úpravy->Vytvořit kontrolní bod*. A vrátit se k předchozímu kontrolnímu bodu pomocí *Úpravy->Vrátit ...*

Funkce kontrolních bodů je doplněk k normálnímu mechanismu Zpět/Znovu při úpravách jednotlivých souborů. Kontrolní body jsou obzvlášť užitečné, když jsou změny rozděleny do více souborů v knize, nebo pokud chcete být schopni vrátit velkou skupinu souvisejících změn jako celku.

Seznam dostupných kontrolních bodů můžete zobrazit pomocí *Zobrazení->Kontrolní body*. Porovnat současný stav knihy se zadaným kontrolním bodem můžete pomocí nástroje *Porovnání e-knih* (stránka 100) – výběrem kontrolního bodu a kliknutím na tlačítko *Porovnat*. Tlačítko *Vrátit* obnoví knihu na vybraný kontrolní bod a zruší všechny změny od vytvoření kontrolního bodu.

1.5.6 Panel Živý náhled


File Preview
Inline formatting
Here, we demonstrate various types of inline text formatting and the use of embedded fonts.
Here is some bold, italic, bold-italic, underlined and struck out text. Then, we have a superscript and a subscript. Now we see some red, green and blue text. Some text with a yellow highlight. Some text in a box. Some text in inverse video.
A paragraph with styled text: subtle emphasis followed by strong text and intense emphasis. This paragraph uses document wide styles for styling rather than inline text properties as demonstrated in the previous paragraph — calibre can handle both with equal ease.
Fun with fonts
This document has embedded the Ubuntu font family. The body text is in the Ubuntu typeface, here is some text in the Ubuntu Mono typeface, notice how every letter has the same width, even i and m. Every embedded font will automatically be embedded in the output ebook during conversion.

Náhled souboru vám poskytuje přehled o různých souborech uvnitř knihy. Panel Živý náhled vám živě ukazuje změny, které provádíte (se zpožděním jedné nebo dvou sekund). Jak upravujete soubory HTML nebo CSS, panel náhledu se automaticky aktualizuje, aby odrážel změny. Když pohybujete kurzorem v editoru, panel náhledu sleduje jeho umístění a ukazuje vám odpovídající umístění v knize. Kliknutí do panelu náhledu způsobí, že kurzor v editoru bude umístěn nad prvek, na který jste kliknuli. Pokud kliknete na odkaz směřující na jiný soubor v knize, bude tento soubor automaticky otevřen panelu úprav a náhledu.

Automatickou synchronizaci pozice a živého náhledu změn můžete vypnout pomocí tlačítek pod panelem náhledu. K živé aktualizaci panelu náhledu dochází pouze tehdy, pokud aktivně nepíšete v editoru, abyste nebyli rušeni nebo zpomalováni čekáním na vykreslení náhledu.

Panel náhledu vám ukazuje, jak bude text vypadat při zobrazení. Panel náhledu však není náhradou za skutečné testování vaší knihy ve skutečném čtecím zařízení. Je jak více, tak méně schopný než skutečná čtečka. Bude tolerovat chyby a nedbalé značení mnohem lépe než většina čtecích zařízení. Také nezobrazí okraje stránek, konce stránek a vložená písma, která používají aliasy názvu písma. Používejte panel náhledu při práci na knize, ale jakmile skončíte, zkontrolujte ji ve skutečném čtecím zařízení nebo softwarovém emulátoru.

Poznámka: Panel náhledu nepodporuje vložená písma, pokud název písma uvnitř souboru písma neodpovídá názvu v pravidle CSS @font-face. Můžete použít nástroj Zkontrolovat knihu pro rychlé nalezení a opravu jakýchkoliv takových problémových písem.

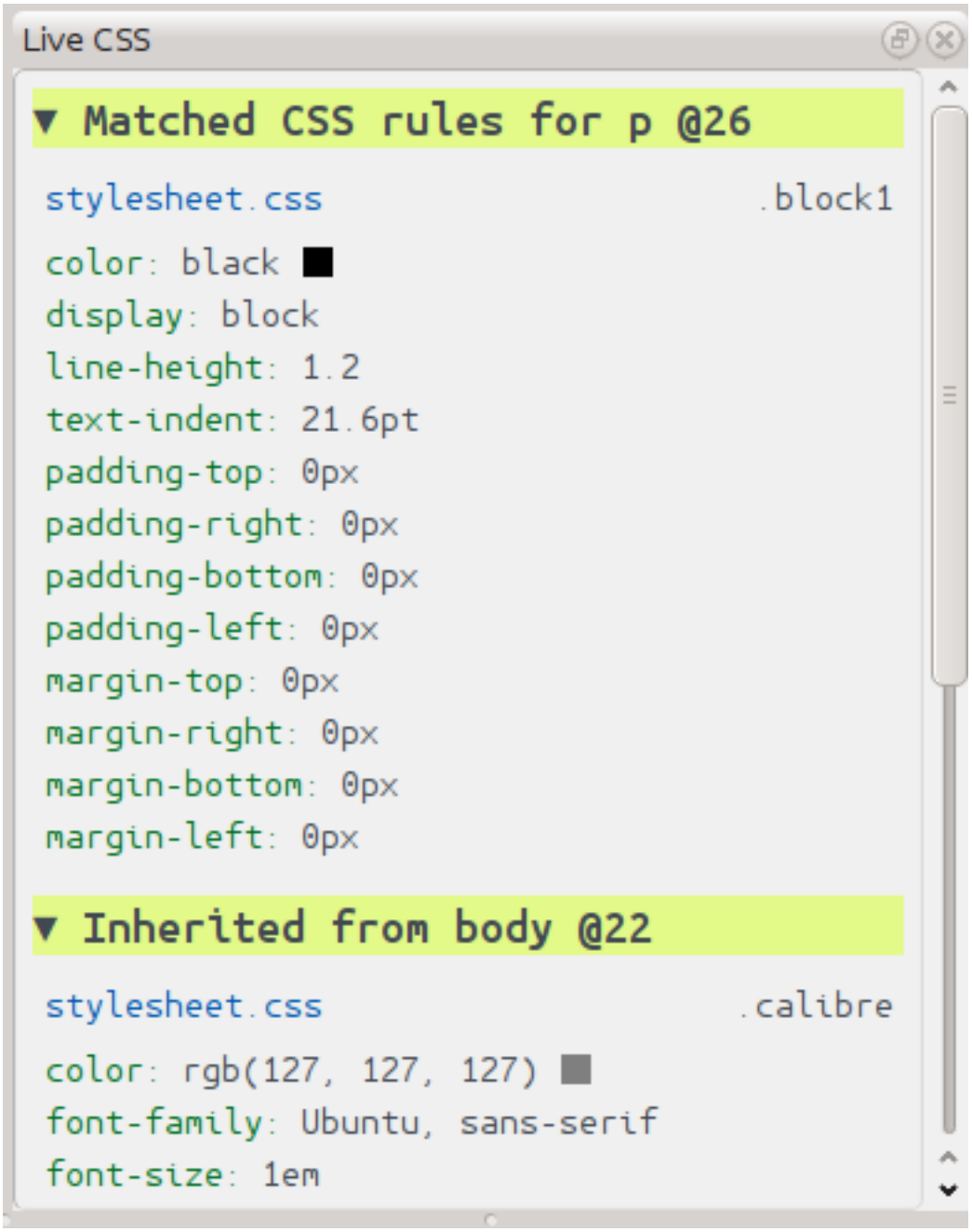
Rozdělování souborů HTML

Jedno možná ne tak zřejmé použití panelu náhledu je rozdělení dlouhých souborů HTML. Během zobrazení souboru, který chcete rozdělit, klikněte na tlačítko *režimu rozdělení* pod panelem náhledu . Pak jednoduše přesuňte myš na místo, kde chcete rozdělit soubor, a klikněte. Tlustá zelená čára vám ukáže, kde přesně dojde k rozdělení, když budete pohybovat myší. Jakmile najdete umístění, které chcete, jednoduše klikněte a rozdělení bude provedeno.

Rozdělení souboru automaticky aktualizuje všechny odkazy, které odkazují na dolní polovinu souboru, a otevře nově rozdělený soubor v editoru.

Jeden soubor HTML můžete také automaticky rozdělit na více místech kliknutím pravým tlačítkem myši uvnitř souboru v editoru a výběrem *Rozdělit na více místech*. To vám umožní snadno rozdělit velký soubor na všech značkách nadpisů nebo všech značkách, které mají určitou třídu, a tak dále.

1.5.7 Panel Živé CSS



Panel *Živé CSS* zobrazí všechna pravidla stylů, která jsou použita na značku, kterou právě upravujete. Je zobrazen název značky společně s jejím číslem řádku v editoru následovaný seznamem odpovídajících pravidel stylů.

Je to skvělý způsob, jak rychle zjistit, která pravidla stylů jsou použita na jakoukoliv značku. Zobrazení má také kliknutelné odkazy (modře), které vás přenesou přímo na místo, kde byl styl definován, pokud chcete provést jakékoliv změny v pravidlech stylu. Jsou zobrazen pravidla stylů, která jsou použita přímo na značku, stejně jako pravidla, která jsou zděděna z nadřazených značek.

Panelu také ukazuje, jaké jsou konečné vypočítané styly pro značku. Vlastnosti v seznamu, které jsou nahrazeny pravidly s vyšší prioritou, jsou zobrazeny přeškrtnuty čárou.

Panel Živé CSS můžete povolit pomocí *Zobrazení->Živé CSS*.

1.5.8 Různé nástroje

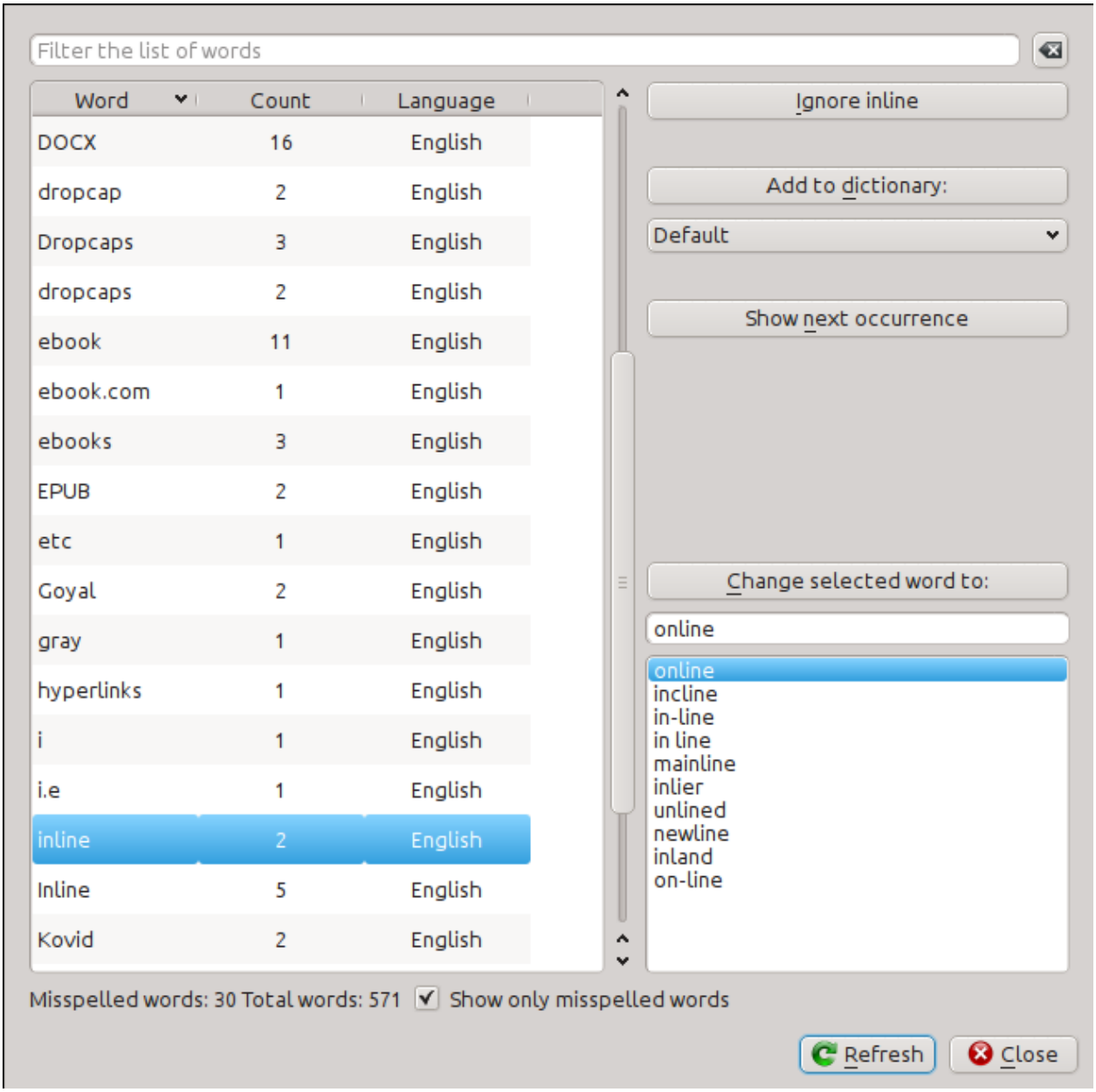
Je zde několik dalších nástrojů, které mohou být užitečné při úpravách knihy.

Zobrazení obsahu

Zobrazení obsahu ukazuje aktuální obsah v knize. Poklepáním na jakoukoliv položku se v editoru otevře místo, na které položka ukazuje. Pro úpravu obsahu, aktualizaci zobrazení nebo rozbalení a sbalení všech položek můžete kliknout pravým tlačítkem myši. Toto zobrazení je přístupné pomocí guilabel:*Zobrazení->Obsah*.

Kontrola pravopisu slov v knize

Kontrolu pravopisu můžete spustit pomocí *Nástroje->Zkontrolovat pravopis*.



Slova jsou zobrazena s počtem výskytů v knize a jazykem, do kterého slovo patří. Informace o jazyku je převzata z metadat knihy a z atributů `lang` v souborech HTML. To umožňuje, aby kontrola pravopisu fungovala dobře i s knihami, které obsahují text ve více jazycích. Například v následujícím úryvku HTML, bude slovo `color` zkontrolováno pomocí americké angličtiny a slovo `colour` pomocí britské angličtiny:

```
<div lang="en_US">color <span lang="en_GB">colour</span></div>
```

Poznámka: Pro zvýraznění dalšího výskytu slova můžete poklepat na toto slovo v editoru. To je užitečné, pokud chcete ručně upravit slovo, nebo vidět, v jakém je v kontextu.

Pro změnu slova jednoduše poklepejte na jeden z nabízených alternativních pravopisů vpravo, nebo napište svůj vlastní opravený pravopis a klikněte na tlačítko *Změnit vybrané slovo na*. To nahradí všechny výskyty slova v knize. Můžete také kliknout pravým tlačítkem myši na slovo v hlavním seznamu slov pro pohodlnou změnu slova z nabídky zobrazené po kliknutí pravým tlačítkem myši.

Můžete nechat kontrolu pravopisu ignorovat slovo pro aktuální relaci kliknutím na tlačítko *Ignorovat*. Můžete také přidat slovo do uživatelského slovníku kliknutím na tlačítko *Přidat slovo do slovníku*. Kontrola pravopisu podporuje více uživatelských slovníků, takže můžete vybrat slovníku, do kterého chcete slovo přidat.

Můžete také nechat kontrolu pravopisu zobrazit všechna slova v knize, ne jen ta špatně napsaná. To je užitečné, abyste viděli, která slova jsou v knize nejčastější a ke spuštění jednoduchého hledání a nahrazení na jednotlivých slovech.

Poznámka: Pokud provedete jakékoliv změny v knize úpravou souborů, když je otevřen nástroj kontroly pravopisu, měli byste kliknout na tlačítko *Aktualizovat* v nástroji kontroly pravopisu. Pokud to neuděláte a budete pokračovat v používání nástroje kontroly pravopisu, mohli byste přijít o změny provedené v editoru.

Přidávání nových slovníků

Kontrola pravopisu obsahuje vestavěné slovníky pro angličtinu a španělštinu. Vlastní slovníky můžete nainstalovat pomocí *Úpravy->Předvolby->Nastavení editoru->Spravovat pravopisné slovníky*. Kontrola pravopisu může použít slovníky z programu LibreOffice (ve formátu .oxt). Tyto slovníky si můžete stáhnout z Repozitáře rozšíření LibreOffice[31].

Vkládání speciálních znaků

Znaky, které jsou obtížně napsat, můžete vložit pomocí nástroje *Úpravy->Vložit speciální znak*. Ten vám zobrazí všechny znaky Unicode, jednoduše klikněte na znak, který chcete zadat. Pokud při kliknutí podržíte klávesu Ctrl, okno se samo zavře po vložení vybraného znaku. Tento nástroj může být použit k vložení speciálních znaků do hlavního textu, nebo do jakékoliv jiné oblasti uživatelského rozhraní, jako je například nástroj Hledat a nahradit.

Protože existuje spousta znaků, můžete definovat své *Oblíbené* znaky, které budou zobrazeny jako první. Jednoduše klikněte pravým tlačítkem myši na znak pro jeho označení jako oblíbeného. Můžete také kliknout pravým tlačítkem myši na znak v oblíbených pro jeho odebrání z oblíbených. A konečně můžete přeuspořádat pořadí znaků v oblíbených kliknutím na tlačítko *Přeuspořádat oblíbené* a přetažením znaku v oblíbených.

Můžete také přímo zadat speciální znaky pomocí klávesnice. Abyste to udělali, zadejte kód Unicode pro znak (šestnáctkově) a pak stiskněte *Alt+X*, což převede dříve zadaný kód na odpovídající znak. Například abyste napsali ÿ, napsali byste ff a pak *Alt+X*. Pro zadání pevné mezery byste použili a0 a pak *Alt+X*, pro zadání vodorovných tří teček byste použili 2026 a *Alt+X* a tak dále.

A nakonec speciální znaky můžete zadat pomocí pojmenovaných entit HTML. Například zadání   bude nahrazeno pevnou mezerou, když napíšete středník. K nahrazení dojde, pouze když napíšete středník.

Zobrazení kontroly kódu

Toto zobrazení vám ukáže kódování HTML a CSS, které jsou použity na aktuální prvek. Otevřete ho kliknutím pravým tlačítkem myši na místo v panelu náhledu a výběrem *Zkontrolovat prvek*. To vám umožní zobrazit kódování HTML pro tento prvek a, co je důležitější, styly CSS, které jsou na něj použity. Můžete dokonce dynamicky upravovat styly a okamžitě uvidíte, jaký vliv mají vaše změny. Pamatujte, že úprava stylů ve skutečnosti neprovádí změny obsahu knihy, umožňuje to pouze rychlé experimentování. Schopnost živé úpravy uvnitř kontroly prvku je ve vývoji.

Kontrola externích odkazů

Tento nástroj můžete použít ke kontrole všech odkazů ve své knize, které odkazují na externí webové stránky. Nástroj se pokusí navštívit každou externě odkazovanou webovou stránku, a pokud návštěva selže, nahlásí všechny poškozené odkazy v pro vás vhodném formátu, abyste je opravili.

[31] http://extensions.libreoffice.org/extension-center?getCategories=Dictionary&getCompatibility=any&sort_on=positive_ratings

Uspořádání souborů do složek podle typu

Při úpravách souborů EPUB, které jste odněkud získali, často zjistíte, že soubory uvnitř EPUB jsou uspořádány nahodile v různých podsložkách. Tento nástroj vám umožňuje automaticky přesunout všechny soubory do podsložek na základě jejich typů. Přístupné je to pomocí *Nástroje->Uspořádat do složek*. Pamatujte, že tento nástroj pouze mění způsob, jakým jsou soubory uspořádány uvnitř EPUB, ale nemění to, jak jsou zobrazeny v Prohlížeči souborů.

Importování souborů v jiných formátech e-knih než EPUB

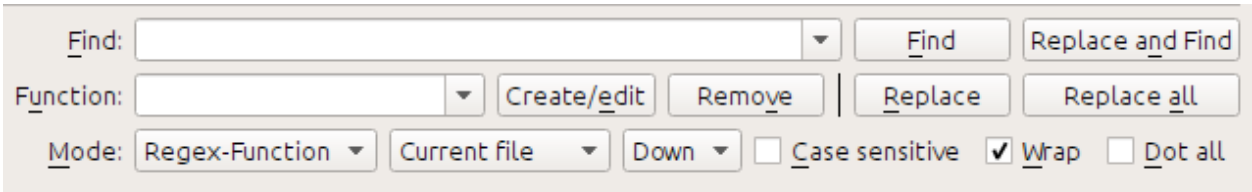
Editor obsahuje schopnost přímo importovat soubory v některých jiných formátech e-knih jako nový EPUB, aniž by prošly úplným převodem. To je obzvlášť užitečné pro přímé vytvoření souborů EPUB z vašich ručně upravených souborů HTML. Můžete to udělat pomocí *Soubor->Importovat soubor HTML nebo DOCX jako novou knihu*.

Function Mode for Search & Replace in the Editor

The Search & Replace tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see Vše o používání regulárních výrazů v Calibre (stránka 163)) with arbitrarily powerful python functions to do all sorts of advanced text processing.

In the standard *regexp* mode for search and replace, you specify both a regular expression to search for as well as a template that is used to replace all found matches. In function mode, instead of using a fixed template, you specify an arbitrary function, in the python programming language[32]. This allows you to do lots of things that are not possible with simple templates.

Techniques for using function mode and the syntax will be described by means of examples, showing you how to create functions to perform progressively more complex tasks.



Automatically fixing the case of headings in the document Here, we will leverage one of the builtin functions in the editor to automatically change the case of all text inside heading tags to title case:

```
Find expression: <([Hh][1-6])[^>]*>.+?</\1>
```

For the function, simply choose the *Title-case text (ignore tags)* builtin function. The will change titles that look like: `<h1>some TITLE</h1>` to `<h1>Some Title</h1>`. It will work even if there are other HTML tags inside the heading tags.

Your first custom function - smartening hyphens The real power of function mode comes from being able to create your own functions to process text in arbitrary ways. The Smarten Punctuation tool in the editor leaves individual hyphens alone, so you can use the this function to replace them with em-dashes.

To create a new function, simply click the Create/Edit button to create a new function and copy the python code from below.

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    return match.group().replace('--', '--').replace('-', '--')
```

[32] https://docs.python.org/2.7/

Every Search & Replace custom function must have a unique name and consist of a python function named replace, that accepts all the arguments shown above. For the moment, we wont worry about all the different arguments to `replace()` function. Just focus on the `match` argument. It represents a match when running a search and replace. Its full documentation in available here[33]. `match.group()` simply returns all the matched text and all we do is replace hyphens in that text with em-dashes, first replacing double hyphens and then single hyphens.

Use this function with the find regular expression:

```
>[^<>]+<
```

And it will replace all hyphens with em-dashes, but only in actual text and not inside HTML tag definitions.

The power of function mode - using a spelling dictionary to fix mis-hyphenated words Often, ebooks created from scans of printed books contain mis-hyphenated words – words that were split at the end of the line on the printed page. We will write a simple function to automatically find and fix such words.

```
import regex
from calibre import replace_entities
from calibre import prepare_string_for_xml

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):

    def replace_word(wmatch):
        # Try to remove the hyphen and replace the words if the resulting
        # hyphen free word is recognized by the dictionary
        without_hyphen = wmatch.group(1) + wmatch.group(2)
        if dictionaries.recognized(without_hyphen):
            return without_hyphen
        return wmatch.group()

    # Search for words split by a hyphen
    text = replace_entities(match.group()[1:-1])  # Handle HTML entities like &amp;
    corrected = regex.sub(r'(\w+)\s*-\s*(\w+)', replace_word, text, flags=regex.VERSION1 | regex.UNI
    return '>%s<' % prepare_string_for_xml(corrected)  # Put back required entities
```

Use this function with the same find expression as before, namely:

```
>[^<>]+<
```

And it will magically fix all mis-hyphenated words in the text of the book. The main trick is to use one of the useful extra arguments to the replace function, `dictionaries`. This refers to the dictionaries the editor itself uses to spell check text in the book. What this function does is look for words separated by a hyphen, remove the hyphen and check if the dictionary recognizes the composite word, if it does, the original words are replaced by the hyphen free composite word.

Note that one limitation of this technique is it will only work for mono-lingual books, because, by default, `dictionaries.recognized()` uses the main language of the book.

Auto numbering sections Now we will see something a little different. Suppose your HTML file has many sections, each with a heading in an `<h2>` tag that looks like `<h2>Some text</h2>`. You can create a custom function that will automatically number these headings with consecutive section numbers, so that they look like `<h2>1. Some text</h2>`.

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    section_number = '%d. ' % number
    return match.group(1) + section_number + match.group(2)
```

[33] https://docs.python.org/2.7/library/re.html#match-objects

```
# Ensure that when running over multiple files, the files are processed
# in the order in which they appear in the book
replace.file_order = 'spine'
```

Use it with the find expression:

```
(?s)(<h2[^<>]*>)(.+?</h2>)
```

Place the cursor at the top of the file and click *Replace all*.

This function uses another of the useful extra arguments to `replace()`: the `number` argument. When doing a *Replace All* number is automatically incremented for every successive match.

Another new feature is the use of `replace.file_order` – setting that to `'spine'` means that if this search is run on multiple HTML files, the files are processed in the order in which they appear in the book. See *Choose file order when running on multiple HTML files* (stránka 95) for details.

Auto create a Table of Contents Finally, lets try something a little more ambitious. Suppose your book has headings in `h1` and `h2` tags that look like `<h1 id="someid">Some Text</h1>`. We will auto-generate an HTML Table of Contents based on these headings. Create the custom function below:

```
from calibre import replace_entities
from calibre.ebooks.oeb.polish.toc import TOC, toc_to_html
from calibre.gui2.tweak_book import current_container
from calibre.ebooks.oeb.base import xml2str

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    if match is None:
        # All matches found, output the resulting Table of Contents.
        # The argument metadata is the metadata of the book being edited
        if 'toc' in data:
            toc = data['toc']
            root = TOC()
            for (file_name, tag_name, anchor, text) in toc:
                parent = root.children[-1] if tag_name == 'h2' and root.children else root
                parent.add(text, file_name, anchor)
            toc = toc_to_html(root, current_container(), 'toc.html', 'Table of Contents for ' + meta
            print (xml2str(toc))
        else:
            print ('No headings to build ToC from found')
    else:
        # Add an entry corresponding to this match to the Table of Contents
        if 'toc' not in data:
            # The entries are stored in the data object, which will persist
            # for all invocations of this function during a 'Replace All' operation
            data['toc'] = []
        tag_name, anchor, text = match.group(1), replace_entities(match.group(2)), replace_entities(m
        data['toc'].append((file_name, tag_name, anchor, text))
        return match.group()  # We don't want to make any actual changes, so return the original mat

# Ensure that we are called once after the last match is found so we can
# output the ToC
replace.call_after_last_match = True
# Ensure that when running over multiple files, this function is called,
# the files are processed in the order in which they appear in the book
replace.file_order = 'spine'
```

And use it with the find expression:

```
<(h[12]) [^<>]* id=['"]([^'"]+)['"][^<>]*>([^<>]+)
```

Run the search on *All text files* and at the end of the search, a window will popup with "Debug Output from your function" which will have the HTML Table of Contents, ready to be pasted into `toc.html`.

The function above is heavily commented, so it should be easy to follow. The key new feature is the use of another useful extra argument to the `replace()` function, the `data` object. The `data` object is a python *dict* that persists between all successive invocations of `replace()` during a single *Replace All* operation.

Another new feature is the use of `call_after_last_match` – setting that to `True` on the `replace()` function means that the editor will call `replace()` one extra time after all matches have been found. For this extra call, the match object will be `None`.

This was just a demonstration to show you the power of function mode, if you really needed to generate a Table of Contents from headings in your book, you would be better off using the dedicated Table of Contents tool in *Tools->Table of Contents*.

The API for the function mode All function mode functions must be python functions named replace, with the following signature:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    return a_string
```

When a find/replace is run, for every match that is found, the `replace()` function will be called, it must return the replacement string for that match. If no replacements are to be done, it should return `match.group()` which is the original string. The various arguments to the `replace()` function are documented below.

The `match` argument The `match` argument represents the currently found match. It is a python Match object[34]. It's most useful method is `group()` which can be used to get the matched text corresponding to individual capture groups in the search regular expression.

The `number` argument The `number` argument is the number of the current match. When you run *Replace All*, every successive match will cause `replace()` to be called with an increasing number. The first match has number 1.

The `file_name` argument This is the filename of the file in which the current match was found. When searching inside marked text, the `file_name` is empty. The `file_name` is in canonical form, a path relative to the root of the book, using `/` as the path separator.

The `metadata` argument This represents the metadata of the current book, such as title, authors, language, etc. It is an object of class *`calibre.ebooks.metadata.book.base.Metadata`* (stránka 155). Useful attributes include, `title`, `authors` (a list of authors) and `language` (the language code).

The `dictionaries` argument This represents the collection of dictionaries used for spell checking the current book. It's most useful method is `dictionaries.recognized(word)` which will return `True` if the passed in word is recognized by the dictionary for the current book's language.

The `data` argument This a simple python `dict`. When you run *Replace All*, every successive match will cause `replace()` to be called with the same `dict` as data. You can thus use it to store arbitrary data between invocations of `replace()` during a *Replace All* operation.

[34] https://docs.python.org/2.7/library/re.html#match-objects

The `functions` argument The `functions` argument gives you access to all other user defined functions. This is useful for code re-use. You can define utility functions in one place and re-use them in all your other functions. For example, suppose you create a function name `My Function` like this:

```
def utility():
    # do something

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    ...
```

Then, in another function, you can access the `utility()` function like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    utility = functions['My Function']['utility']
    ...
```

You can also use the functions object to store persistent data, that can be re-used by other functions. For example, you could have one function that when run with *Replace All* collects some data and another function that uses it when it is run afterwards. Consider the following two functions:

```
# Function One
persistent_data = {}

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    ...
    persistent_data['something'] = 'some data'

# Function Two
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    persistent_data = functions['Function One']['persistent_data']
    ...
```

Debugging your functions You can debug the functions you create by using the standard `print()` function from python. The output of print will be displayed in a popup window after the Find/replace has completed. You saw an example of using `print()` to output an entire table of contents above.

Choose file order when running on multiple HTML files When you run a *Replace All* on multiple HTML files, the order in which the files are processes depends on what files you have open for editing. You can force the search to process files in the order in which the appear by setting the `file_order` attribute on your function, like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    ...

replace.file_order = 'spine'
```

`file_order` accepts two values, `spine` and `spine-reverse` which cause the search to process multiple files in the order they appear in the book, either forwards or backwards, respectively.

Having your function called an extra time after the last match is found Sometimes, as in the auto generate table of contents example above, it is useful to have your function called an extra time after the last match is found. You can do this by setting the `call_after_last_match` attribute on your function, like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args, **kwargs):
    ...

replace.call_after_last_match = True
```

Fragmenty

The calibre editor supports *snippets*. A snippet is a piece of text that is either re-used often or contains a lot of redundant text. The editor allows you to insert a snippet with only a few key strokes. For example, suppose you often find yourself inserting link tags when editing HTML files, then you can simply type `<a` in the editor and press `Control+J`. The editor will expand it to:

```
<a href="filename"></a>
```

Not only that, the word `filename` will be selected, with the cursor placed over it, so that you can easily type in the real filename, using the editor's nifty *Automatické dokončování* (stránka 100) feature. And once you are done typing the filename, press `Control+J` again and the cursor will jump to the position in between the `<a>` tags so you can easily type in the text for the link.

The snippets system in the editor is very sophisticated, there are a few built-in snippets and you can create your own to suit your editing style.

The following discussion of the built-in snippets should help illustrate the power of the snippets system.

Poznámka: You can also use snippets in the text entry fields in the Search and Replace panel, however, placeholders (using `Control+J` to jump around) will not work.

The built-in snippets The built-in snippets are described below. Note that you can override them by creating your own snippets with the same trigger text.

Inserting filler text [Lorem] The first built-in snippet, and the simplest is used to insert filler text into a document. The filler text is taken from De finibus bonorum et malorum[35] a philosophical work by Cicero (translated to English). To use it simply type `Lorem` in an HTML file and press `Control+J`. It will be replaced by a couple of paragraphs of filler.

The definition of this snippet is very simple, the trigger text is defined as `Lorem` and the template is defined simply as the literal text to be inserted. You can easily customize it to use your favorite form of filler text.

Inserting a self-closing HTML tag [<>] Now let's look at a simple example of the powerful concept of *placeholders*. Say you want to insert the self-closing tag `<hr/>`. Just type `<>`, and press `Control+J`, the editor will expand the snippet to:

```
<|/>
```

Here, the `|` symbol represents the current cursor position. You can then type `hr` and press `Control+J` to move the cursor to after the end of the tag. This snippet is defined as:

```
Trigger: <>
Template: <$1/>$2
```

Placeholders are simply the dollar ($) sign followed by a number. When the snippet is expanded by pressing `Control+J` the cursor is positioned at the first placeholder (the placeholder with the lowest number). When you press `Control+J` again the cursor jumps to the next placeholder (the placeholder with the next higher number).

Inserting an HTML link tag [<a] HTML link tags all share a common structure. They have an `href` attribute and some text between the opening and closing tags. A snippet to make typing them more efficient will introduce us

[35] http://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum

to some more features of placeholders. To use this snippet, simply type <a and press Control+J. The editor will expand this to:

```
<a href="filename|"></a>
```

Not only that, the word filename will be selected, with the cursor placed over it, so that you can easily type in the real filename, using the editor's nifty *Automatické dokončování* (stránka 100) feature. And once you are done typing the filename, press Control+J again and the cursor will jump to the position in between the <a> tags so you can easily type in the text for the link. After you are done typing the text, press Control+J again to jump to the point after the closing tag. This snippet is defined as:

```
Trigger: <a
Template: <a href="${1:filename}">${2*}</a>$3
```

There are a couple of new features here. First the $1 placeholder has become more complex. It now includes some *default text* (the word filename. If a placeholder contains default text, the default text is substituted for the placeholder when the snippet is expanded. Also when you jump to a placeholder with default text using Control+J, the default text is selected. In this way, you can use default text to act as a reminder to you to fill in important parts of the template. You can specify default text for a placeholder by using the syntax: ${<number>:default text}.

The other new feature is that the second placeholder has an asterisk after it (${2*}). This means that any text that was selected before expanding the template is substituted for the placeholder. To see this in action, select some text in the editor, press Control+J, type <a and press Control+J again, the template will be expanded to:

```
<a href="filename">whatever text you selected</a>
```

Inserting a HTML image tag [<i] This is very similar to inserting an HTML link, as we saw above. It allows you to quickly input an tag and jump between the src and alt attributes:

```
Trigger: <i
Template: <img src="${1:filename}" alt="${2*:description}" />$3
```

Insert an arbitrary HTML tag [<<] This allows you to insert an arbitrary full HTML tag (or wrap previously selected text in the tag). To use it, simply type << and press Control+J.The editor will expand it to:

```
<|></>
```

Type the tag name, for example: span and press Control+J, that will result in:

```
<span>|</span>
```

You will note that the closing tag has been automatically filled with span. This is achieved with yet another feature of placeholders, *mirroring*. Mirroring simply means that if you specify the sample placeholder more than once in a template, the second and all later positions will be automatically filled in with whatever you type in the first position, when you press Control+J. The definition for this snippet is:

```
Trigger: <<
Template: <$1>${2*}</$1>$3
```

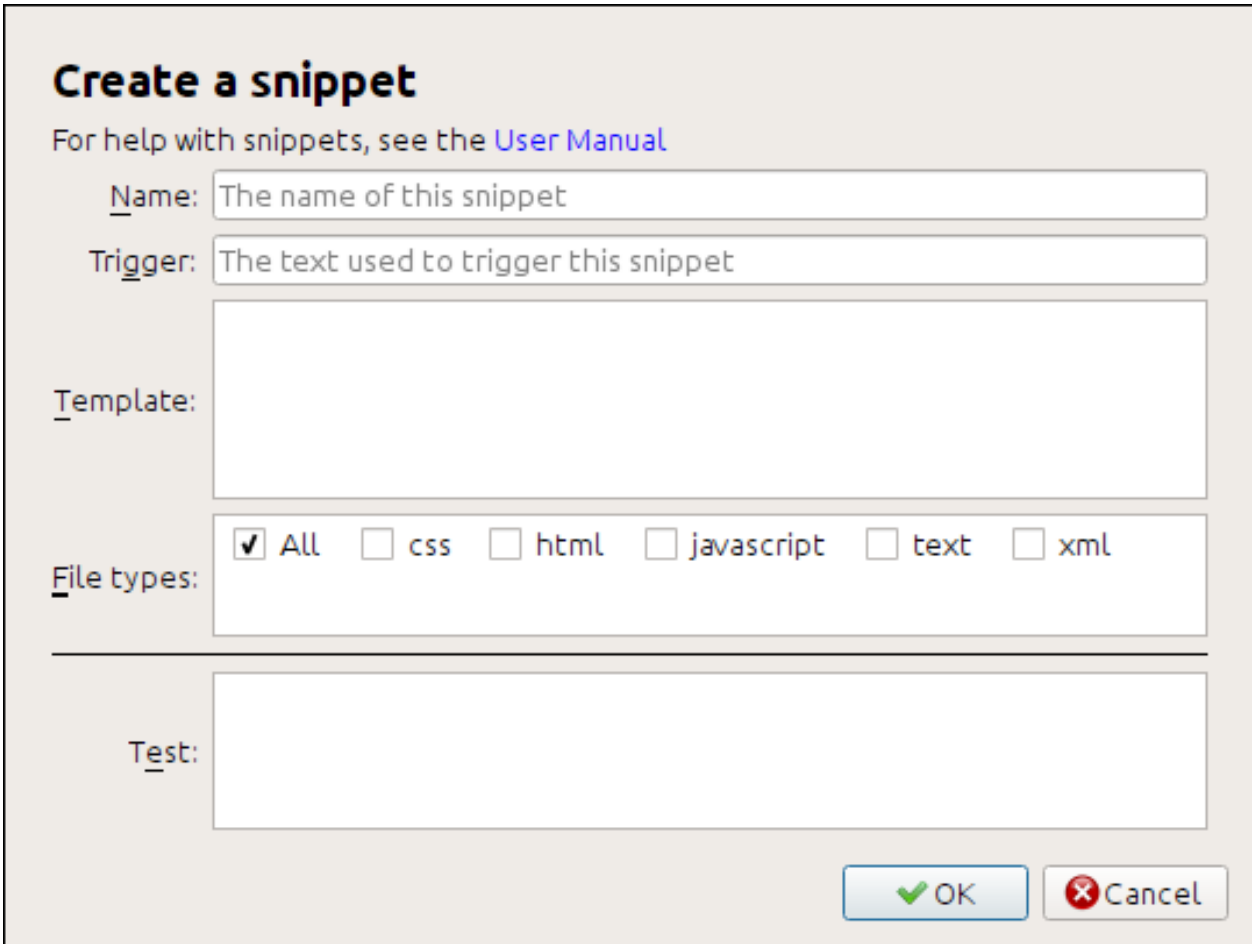
As you can see, the first placeholder ($1) has been specified twice, the second time in the closing tag, which will simply copy whatever you type in the opening tag.

Inserting an arbitrary HTML tag with a class attribute [<c] This is very similar to the insert arbitrary tag example above, except that it assumes that you want to specify a class for the tag:

```
Trigger: <c
Template: <$1 class="${2:classname}">${3*}</$1>$4
```

This will allow you to first type the tag name, press `Control+J`, type the class name, press `Control+J` type the contents of the tag and press `Control+J` one last time to jump out of the tag. The closing tag will be auto-filled.

Creating your own snippets Snippets really shine because you can create your own to suit your editing style. To create your own snippets go to *Edit->Preferences->Editor settings->Manage snippets* in the editor. This will pop-up an easy to use dialog to help you create your own snippets. Simply click the *Add snippet* button and you will see a dialog that looks like:



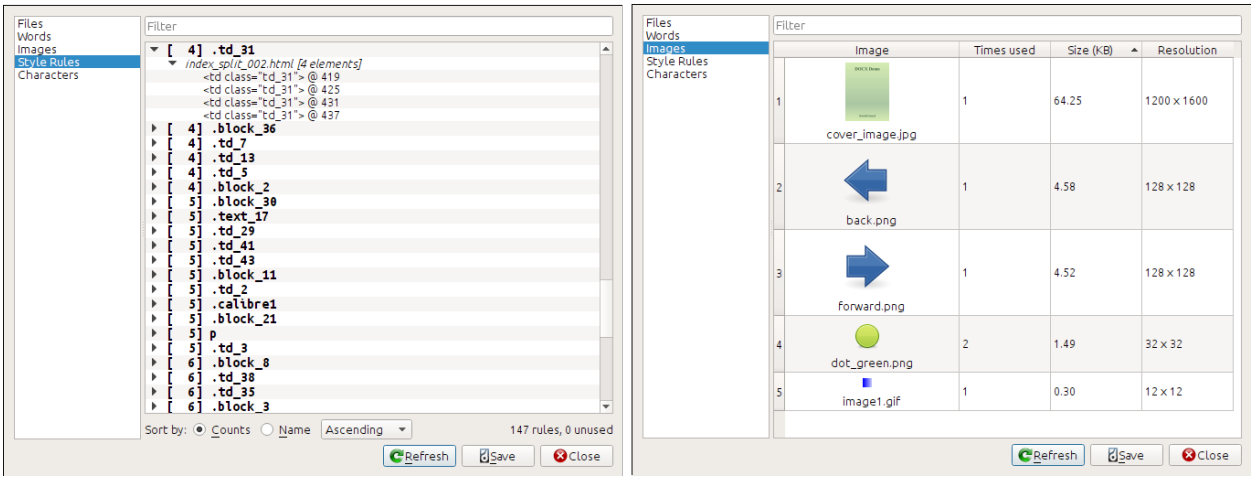
First give your snippet a name, something descriptive, to help identify the snippet in the future. Then specify the *trigger*. A trigger is simply the text that you have to type in the editor before pressing `Control+J` in order to expand the snippet.

Then specify the snippet template. You should start with one of the example above and modify it to suit your needs. Finally, specify which file types you want the snippet to be active for. This way you can have multiple snippets with the same trigger text that work differently in different file types.

The next step is to test your newly created snippet. Use the *Test* box at the bottom. Type in the trigger text and press `Control+J` to expand the snippet and jump between placeholders.

Nástroj Hlášení

Editor obsahuje pěkný nástroj *Hlášení* (přes *Nástroje->Hlášení*), který zobrazuje souhrny souborů, obrázků, odkazů, slov, znaků a stylů použitých v knize. Každý řádek v hlášení je propojený. Poklepáním na řádek skočíte na místo v knize, kde je tato položka použita nebo definována (podle potřeby). Například v zobrazení *Odkazy* můžete poklepat na položky sloupce *Zdroj*, abyste skočili tam, kde je odkaz definován, a na položky ve sloupci *Cíl*, abyste skočili, kam odkaz ukazuje.



1.5.9 Speciální funkce v editoru kódu

Editor HTML v Calibre je velice výkonný. Má mnoho funkcí, díky kterým je úprava HTML (a CSS) jednodušší.

Zvýraznění syntaxe

Editor HTML má velice propracované zvýraznění syntaxe. Funkce zahrnují:

- Text uvnitř značek pro tučné písmo, kurzívu a nadpisy jsou tučné nebo kurzívou
- Během pohybu kurzorem přes HTML jsou odpovídající značky HTML zvýrazněny
- Neplatné HTML je zvýrazněno červeným podtržením
- Pravopisné chyby v textu uvnitř značek HTML a atributy, jako jsou názvy, jsou zvýrazněny. Kontrola pravopisu je schopná rozpoznat jazyk na základě hodnoty atributu `lang` aktuální značky a celkového jazyka knihy.
- CSS vložené uvnitř značek `<style>` je zvýrazněno
- Speciální znaky, které může být těžké rozlišit, jako jsou pevné mezery, různé druhy pomlček atd., jsou zvýrazněny.
- Odkazy na jiné soubory ve značkách `<a>`, `<img>` a `<link>` mají všechny zvýrazněné názvy souborů. Pokud odkazují na název souboru, který neexistuje, je název souboru označen červeným podtržením.

Nápověda závislá na kontextu

Pro získání nápovědy ke značce HTML nebo vlastnosti CSS můžete kliknout pravým tlačítkem myši na název této značky nebo vlastnosti.

Můžete také podržet klávesu Ctrl a kliknout na jakýkoliv název souboru uvnitř značky odkazu pro automatické otevření tohoto souboru v editoru.

Automatické dokončování

Při úpravě e-knihy je jedním z nejobtížnějších úkolů vytvořit odkazy na jiné soubory uvnitř knihy, šablony stylů CSS nebo obrázky. Musíte vyřešit správný název souboru a relativní cestu k souboru. Editor má automatické dokončování, aby toto zjednodušil.

Během psaní názvu souboru editor automaticky nabízí návrhy. Jednoduše použijte klávesu Tab pro výběr správného názvu souboru. Editor dokonce nabízí návrhy pro odkazy ukazující na kotvu uvnitř jiného souboru HTML. Po napsání znaku `#` zobrazí editor seznam všech kotev v cílovém souboru s malým úryvkem textu, aby vám pomohl zvolit správnou kotvu.

Pamatujte, že na rozdíl od většiny ostatních systémů dokončování, systém dokončování editoru používá podposloupnou shodu. To znamená, že pro dokončení názvu souboru můžete zadat jen dvě nebo tři písmena odkudkoliv z názvu souboru . Řekněme například, že když chcete název souboru `../images/arrow1.png`, můžete jednoduše napsat `ia1` a stisknout Tab pro dokončení názvu souboru. Při hledání shod systém dokončování upřednostňuje písmena, která jsou na začátku slova, nebo ihned po oddělovači cesty. Jakmile si zvyknete na tento systém, zjistíte, že vám ušetří spoustu času a úsilí.

Fragmenty

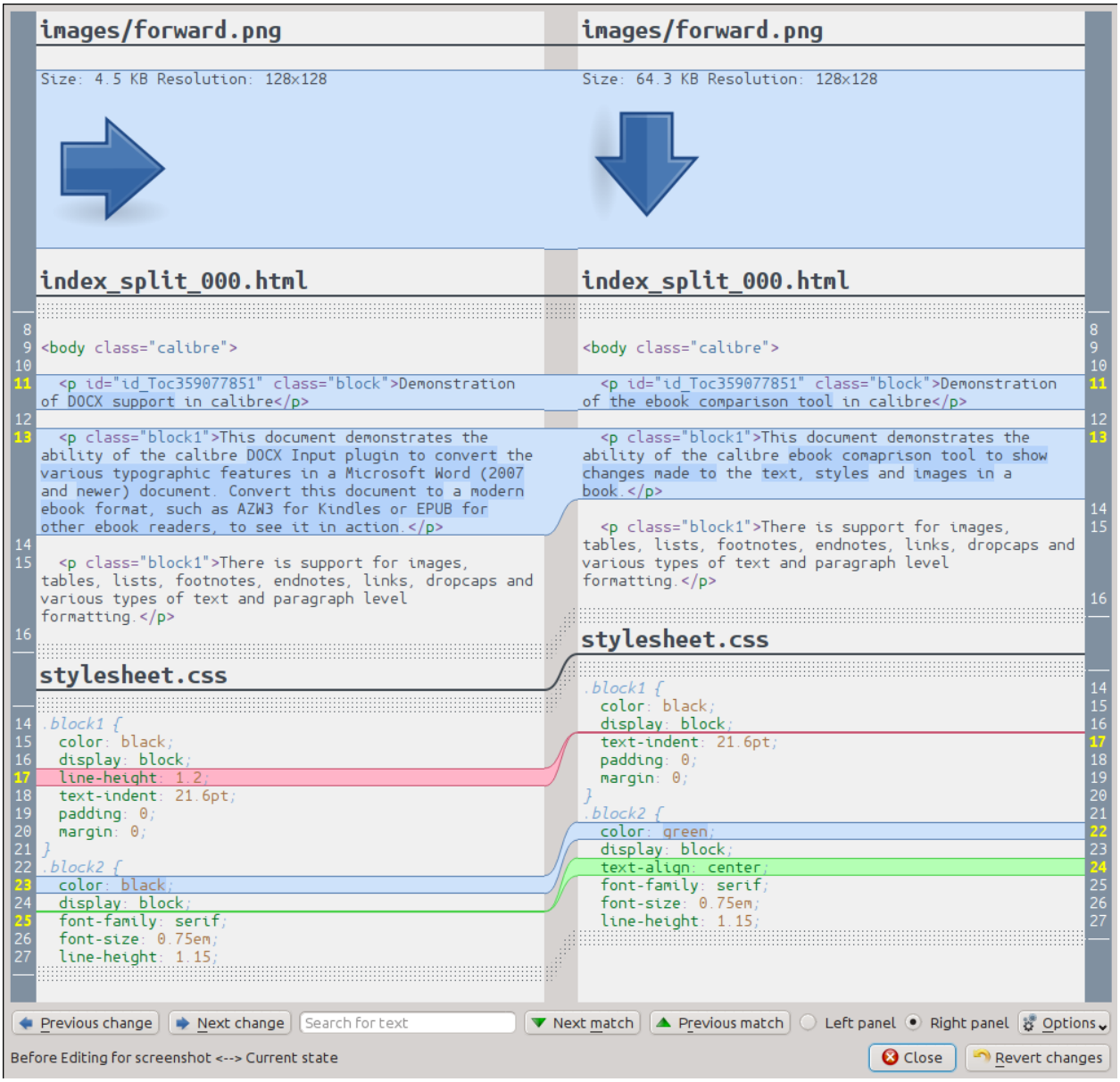
Editor Calibre podporuje *fragmenty*. Fragment je kus textu, který je buď často znovu použit nebo obsahuje hodně nadbytečného textu. Editor vám umožňuje vložit fragment pomocí jen několika stisků kláves. Fragmenty jsou velice výkonné, s mnoha funkcemi, jako jsou zástupné symboly, mezi kterými můžete přeskakovat, automatické zrcadlení opakované textu a tak dále. Pro více informací navštivte Fragmenty (stránka 96).

1.6 Porovnání e-knih

calibre zahrnuje integrovaný nástroj porovnání e-knih, který může být použit pro zobrazení, co se změnilo uvnitř e-knihy po úpravě nebo převodu. Umí porovnat knihy ve formátech EPUB a AZW3.

Pro jeho použití buď otevřete e-knihu v nástroji pro *Úprava e-knih* (stránka 71) a klikněte na *Soubor->Porovnat s jinou knihou* nebo použijte panel *Podrobnosti o knize* (stránka 16). Pokud provádíte převod z EPUB do EPUB, původní soubor EPUB bude uložen jako ORIGINAL_EPUB. Jednoduše klikněte pravým tlačítkem myši na položku ORIGINAL_EPUB v panelu Podrobnosti o knize a zvolte *Porovnat s formátem EPUB*.

Nástroj porovnání, který se otevře, bude vypadat jako snímek obrazovky níže. Ukáže vám rozdíly v textu, stylech a obrázcích ve zvolených knihách.


images/forward.png
Size: 4.5 KB Resolution: 128x128
images/forward.png
Size: 64.3 KB Resolution: 128x128
index_split_000.html
<body class="calibre">
Demonstration of DOCX support in calibre
This document demonstrates the ability of the calibre DOCX Input plugin to convert the various typographic features in a Microsoft Word (2007 and newer) document. Convert this document to a modern ebook format, such as AZW3 for Kindles or EPUB for other ebook readers, to see it in action.
There is support for images, tables, lists, footnotes, endnotes, links, dropcaps and various types of text and paragraph level formatting.
index_split_000.html
<body class="calibre">
Demonstration of the ebook comparison tool in calibre
This document demonstrates the ability of the calibre ebook comaprison tool to show changes made to the text, styles and images in a book.
There is support for images, tables, lists, footnotes, endnotes, links, dropcaps and various types of text and paragraph level formatting.
stylesheet.css
.block1 {
color: black;
display: block;
line-height: 1.2;
text-indent: 21.6pt;
padding: 0;
margin: 0;
}
.block2 {
color: black;
display: block;
font-family: serif;
font-size: 0.75em;
line-height: 1.15;
stylesheet.css
.block1 {
color: black;
display: block;
text-indent: 21.6pt;
padding: 0;
margin: 0;
}
.block2 {
color: green;
display: block;
text-align: center;
font-family: serif;
font-size: 0.75em;
line-height: 1.15;
Previous change
Next change
Search for text
Next match
Previous match
Left panel
Right panel
Options
Before Editing for screenshot <--> Current state
Close
Revert changes

1.6.1 Pochopení zobrazení porovnání

Jak můžete vidět na snímku obrazovky výše, zobrazení porovnání ukazuje rozdíly mezi dvěma knihami vedle sebe. Jsou zobrazeny pouze rozdíly s několika řádky kontextu kolem nich. Tak můžete snadno vidět na první pohled pouze to, co se změnilo uvnitř velkého dokumentu, jako je kniha.

Přidaný text je zobrazen se zeleným pozadím, odebraný text s červeným pozadím a změněný text s modrým pozadím.

Čísla řádků všeho změněného textu jsou zobrazena po stranách, umožňují snadno přejít na konkrétní změnu v editoru. Když otevřete nástroj porovnání z editoru, můžete také poklepat na řádek v pravém panelu pro automatické přejití na tento řádek v editoru.

Užitečná technika při porovnávání knih je říct nástroji porovnání, aby zkrášlil textové soubory a soubory stylů, než vypočítá rozdíly. To může mít často za výsledek čistější a snadněji sledovatelné rozdíly. Abyste to udělali, klikněte na tlačítko *Volby* vpravo dole a zvolte *Zkrášlit soubory před jejich porovnáním*. Pamatujte, že zkrášlení může mít někdy

nežádoucí účinky, protože může způsobit, že jsou neplatné značky změněny, aby byly platné. Pomocí tlačítka *Volby* můžete také změnit počet řádků kontextu zobrazovaného kolem rozdílů.

Pomocí panelu hledání v dolní části můžete hledat jakýkoliv text v rozdílech. Budete muset zadat, ve kterém panelu hledat, *Levý* nebo *Pravý*.

1.6.2 Jak spustit nástroj porovnání

Nástroj porovnání je nejužitečnější, když máte dvě verze stejné knihy a chcete vidět, jaký je mezi nimi rozdíl. Za tímto účelem je několik způsobů spuštění nástroje.

Porovnání dvou souborů e-knih

Otevřete první soubor v nástroji *Úprava e-knih* (stránka 71). Nyní klikněte na *Soubor->Porovnat s jinou knihou* a zvolte druhý soubor (musí být ve stejném formátu jako první). Otevře se zobrazení porovnání s upravovaným souborem vpravo a druhým souborem vlevo.

Porovnání ORIGINAL_FMT s FMT

Když provádíte v calibre převod z FMT do něho samého, původní soubor je uložen jako ORIGINAL_FMT. Co se změnilo převodem, můžete zobrazit kliknutím pravým tlačítkem myši na položku ORIGINAL_FMT v panelu *Podrobnosti o knize* (stránka 16) v hlavním okně calibre a výběrem *Porovnat s FMT*. Otevře se zobrazení porovnání s ORIGINAL_FMT vlevo a FMT vpravo.

Porovnání kontrolního bodu s aktuálním stavem knihy během úpravy

Nástroj *Úprava e-knih* (stránka 71) má velice užitečnou funkci zvanou *Kontrolní body* (stránka 83). Ta vám umožňuje uložit aktuální stav knihy jako pojmenovaný *kontrolní bod*, do kterého se můžete vrátit, pokud se vám nelíbí změny, které jste udělali od vytvoření kontrolního bodu. Kontrolní body jsou také vytvářeny automaticky, když provedete různé automatizované akce v editoru. Seznam kontrolních bodů můžete vidět přejitím na *Zobrazení->Kontrolní body* a pak použitím tlačítka *Porovnat* pro porovnání knihy ve vybraném kontrolním bodu s aktuálním stavem. Nástroj porovnání zobrazí kontrolní bod vlevo a aktuální stav vpravo.

1.7 Úprava metadat e-knihy

Obsah

- *Úprava metadat jedné knihy najednou* (stránka 103)
  - *Stahování metadat* (stránka 103)
  - *Správa formátů knihy* (stránka 103)
  - *Vše o obálkách* (stránka 103)
- *Úprava metadat mnoha knih najednou* (stránka 103)
  - *Hledat a nahradit* (stránka 104)
  - *Hromadné stahování metadat* (stránka 105)

E-knihy se objevují ve všech tvarech a velikostech a spíše častěji než ne jsou jejich metadata (věci, jako jsou název, autor, série, vydavatel) neúplná nebo nesprávná. Nejjednodušší způsob, jak změnit metadata v calibre, je jednoduše poklepat na položku a zadat správné nahrazení. Pro propracovanější "výkonné úpravy" použijte nástroje pro úpravu metadat probrané níže.

1.7.1 Úprava metadat jedné knihy najednou

Klikněte na knihu, kterou chcete upravit, a pak klikněte na tlačítko *Upravit metadata* nebo stiskněte klávesu E. Otevře se dialogové okno, které vám umožní upravit všechny aspekty metadat. Má různé funkce, aby byly úpravy rychlejší a efektivnější. Seznam běžně používaných tipů:

- Můžete kliknout na tlačítko mezi názvem a autory pro jejich automatické prohození.
- Můžete kliknout na tlačítko vedle řazení autora, aby ho calibre automaticky vyplnilo pomocí hodnot řazení uložených u každého autora. Použije dialogové okno *Spravovat autory*, abyste zobrazili a změnili hodnoty řazení autorů. Toto dialogové okno lze otevřít kliknutím a podržením tlačítka vedle řazení autora.
- Můžete kliknout na tlačítko vedle štítků pro použití Editoru štítků pro správu štítků přidružených ke knize.
- Pole ISBN bude mít červené pozadí, pokud zadáte neplatné ISBN. Pro platná ISBN bude zelené.
- Pole řazení autora bude červené, pokud se hodnota řazení autora liší od té, kterou calibre předpokládá.

Stahování metadat

Nejhezčí funkce dialogového okna úpravy metadat je jeho schopnost automaticky vyplnit mnoho polí metadat získáním metadat z různých webových stránek. calibre aktuálně používá isbndb.com, Google Books, Amazon a Library Thing. Stažená metadata za vás mohou vyplnit název, autora, sérii, štítky, hodnocení, popis a ISBN.

Pro použití stahování vyplňte pole název a autor a klikněte na tlačítko *Stáhnout metadata*. calibre vám nabídne seznam knih, které nejvíce odpovídají názvu a autorovi. Pokud vyplníte nejdříve pole ISBN, bude použito přednostně před názvem a autorem. Pokud nejsou nalezeny žádné shody, zkuste učinit hledání trochu méně konkrétní zahrnutím pouze některých klíčových slov v názvu a použitím pouze příjmení autora.

Správa formátů knihy

V calibre může mít jedna kniha přidruženo mnoho různých *formátů*. Například jste mohli získat kompletní shakespearovo dílo ve formátu EPUB a později ho převést do MOBI, abyste ho mohli číst na Kindle. calibre pro vás automaticky spravuje více formátů. V oddílu *Dostupné formáty* dialogového okna Upravit metadata můžete tyto formáty spravovat. Můžete přidat nový formát, odstranit existující formát a také požádat calibre o nastavení metadat a obálky pro položku knihy z metadat v jednom z formátů.

Vše o obálkách

Můžete požádat calibre, aby pro vás stáhlo obálky knih, za předpokladu, že kniha má známé ISBN. Případně můžete zadat soubor ve svém počítači, který chcete použít jako obálku. calibre pro vás může dokonce vygenerovat výchozí obálku se základními metadaty. Můžete přetahovat obrázky na obálku, abyste ji změnili, a také kliknutím pravým tlačítkem myši kopírovat a vložit obrázky obálky.

Kromě toho tu je tlačítko pro automatické oříznutí rámečků obálky v případě, že má vaše obálka ošklivý rámeček.

1.7.2 Úprava metadat mnoha knih najednou

Nejdříve vyberte knihy, které chcete upravit, podržením Ctrl nebo Shift a kliknutím na ně. Pokud vyberete více než jednu knihu, kliknutí na tlačítko *Upravit metadata* způsobit otevření nového dialogového okna *hromadné* úpravy metadat. Pomocí tohoto dialogového okna můžete rychle nastavit autora, vydavatele, hodnocení, štítky, sérii atd. mnoha knih na stejnou hodnotu. To je obzvlášť užitečné, pokud jste právě importovali řadu knih, které mají některá společná metadata. Toto dialogové okno je velice výkonné, například má kartu Hledat a nahradit, kterou můžete použít k provedení hromadných operací na metadatech a dokonce i ke kopírování metadat z jednoho sloupce do jiného.

Normální dialogové okno úpravy metadat má také tlačítka Další a Předchozí, která můžete použít k úpravě metadat několika knih jednu po druhé.

Hledat a nahradit

Dialogové okno hromadné úpravy metadat vám umožňuje provádět libovolně výkonné operace hledání a nahrazení na vybraných knihách. Ve výchozím nastavení používá jednoduché textové hledání a nahrazení, ale také podporuje *regulární výrazy*. Pro více informací o regulárních výrazech se podívejte na *Vše o používání regulárních výrazů v Calibre* (stránka 163).

Jak již bylo zmíněno výše, existují dva režimy hledání a nahrazení: znaková shoda a regulární výraz. Znaková shoda bude hledat ve *Vyhledávacím poli*, které zvolíte, znaky, které zadáte do pole *Hledat*, a nahradí tyto znaky tím, co zadáte do pole *Nahradit čím*. Bude nahrazen každý výskyt hledaných znaků. Předpokládejme například, že vyhledávací pole obsahuje *o zlobivé kočičce*. Pokud hledáte *o*, které má být nahrazeno *AHOJ*, pak výsledek bude *AHOJ zlAHOJbivé kAHOJčičce*.

Pokud pole, ve kterém hledáte, je *vícenásobné* pole, jako jsou štítky, pak se každý štítek posuzuje samostatně. Například pokud vaše štítky obsahují *Horor, Strašidelné*, vyhledávací výraz *r,* nebude odpovídat ničemu, protože výraz bude nejdříve použit na *Horor* a pak na *Strašidelné*.

Pokud chcete, aby hledání ignorovalo rozdíly mezi velkými a malými písmeny, zrušte zaškrtnutí pole *Rozlišovat velikost písmen*.

Můžete nechat calibre, aby změnilo velikost písmen výsledku (informace po provedení nahrazení) výběrem jedné z funkcí z pole *Použít funkci po nahrazení*. Dostupné operace jsou:

- *Malá písmena* – změnit všechny znaky v poli na malá písmena
- *Velká písmena* – změnit všechny znaky v poli na velká písmena
- *Všechna první velká* – převést každé slovo ve výsledku na velká počáteční písmena

Pole *Váš test* je vám k dispozici pro zadání textu, abyste si ověřili, že hledání a nahrazení dělá to, co chcete. Ve většině případů budou stačit testovací pole knih, ale je možné, že existuje případ, který chcete zkontrolovat, a který není zobrazen v těchto polích. Zadejte tento případ do *Váš test*.

Režim regulární výraz má některé rozdíly od znakového režimu, kromě (samozřejmě) používání regulárních výrazů. První je, že funkce jsou použity na části řetězce odpovídající hledanému řetězci, nikoliv na celá pole. Druhý je, že se funkce použijí na řetězec nahrazení, nikoliv na celé pole.

Třetí a nejdůležitější je, že řetězec nahrazení může odkazovat na části hledaného řetězce pomocí zpětných odkazů. Zpětný odkaz je `\\n`, kde n je celé číslo, které odkazuje na ntou skupinu v závorkách ve vyhledávacím výrazu. Například, se stejným příkladem jako výše, *o zlobivé kočičce*, vyhledávací výraz *o (...) (...)* a výraz nahrazení *o \2 \1*, bude výsledek *o kočičce zlobivé*. Pro více informací o zpětných odkazech se podívejte na *Vše o používání regulárních výrazů v Calibre* (stránka 163).

Jeden užitečný vzor: předpokládejme, že chcete změnit velikost písmen celého pole. Nejsnadnější způsob, jak to udělat, je použít znakový režim, ale předpokládejme dále, že chcete použít režim regulárního výrazu. Vyhledávací výraz by měl být *(.*)*, výraz nahrazení by měl být *\1* a měla by být vybrána požadovaná funkce změny velikosti písmen.

A konečně, v režimu regulárního výrazu můžete kopírovat hodnoty z jednoho pole do druhého. Stačí, aby se zdrojové a cílová pole lišilo. Kopie může nahradit cílové pole, předřadit se poli (přidat na začátek) nebo se připojit k poli (přidat na konec). Zaškrtávací políčko 'použít čárku' říká calibre, aby přidalo (nebo nepřidalo) čárku mezi text a cílové pole v režimech předřazení a připojení. Pokud je cíl vícenásobný (např. štítky), pak nemůžete zrušit zaškrtnutí tohoto políčka.

Hledat a nahradit je provedeno po použití všech ostatních změn metadat na ostatních kartách. To může vést k určitému zmatku, protože testovací pole zobrazí informace před ostatními změnami, ale operace bude použita po ostatních změn. Pokud máte jakékoliv pochybnosti o tom, co se stane, nemíchejte hledání a nahrazení s ostatními změnami.

Hromadné stahování metadat

Pokud chcete stáhnout metadata pro více knih najednou, klikněte pravým tlačítkem myši na tlačítko *Upravit metadata* a vyberte *Stáhnout metadata*. Můžete si zvolit stáhnout pouze metadata, pouze obálky nebo obojí.

1.8 Nejčastější dotazy

Obsah

- *Ebook Format Conversion* (stránka 105)
- *Device Integration* (stránka 108)
- *Library Management* (stránka 115)
- *Různé* (stránka 119)

1.8.1 Ebook Format Conversion

Obsah

- *What formats does calibre support conversion to/from?* (stránka 105)
- *Jaké jsou nejlepší zdrojové formáty pro převod?* (stránka 106)
- *Převedl jsem PDF soubor, ale výsledný soubor má různé problémy?* (stránka 106)
- *How do I convert my file containing non-English characters, or smart quotes?* (stránka 106)
- *What's the deal with Table of Contents in MOBI files?* (stránka 106)
- *The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.* (stránka 107)
- *Jak převedu kolekci HTML souborů v určitém pořadí?* (stránka 107)
- *The EPUB I produced with calibre is not valid?* (stránka 108)
- *How do I use some of the advanced features of the conversion tools?* (stránka 108)

What formats does calibre support conversion to/from?

calibre supports the conversion of many input formats to many output formats. It can convert every input format in the following list, to every output format.

Input Formats: AZW, AZW3, AZW4, CBZ, CBR, CBC, CHM, DJVU, DOCX, EPUB, FB2, HTML, HTMLZ, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ

Output Formats: AZW3, EPUB, DOCX, FB2, HTMLZ, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

Poznámka: PRC is a generic format, calibre supports PRC files with TextRead and MOBIBook headers. PDB is also a generic format. calibre supports eReader, Plucker, PML and zTxt PDB files. DJVU support is only for converting DJVU files that contain embedded text. These are typically generated by OCR software. MOBI books can be of two types Mobi6 and KF8. calibre fully supports both. MOBI files often have .azw or .azw3 file extensions. DOCX files from Microsoft Word 2007 and newer are supported.

Jaké jsou nejlepší zdrojové formáty pro převod?

In order of decreasing preference: LIT, MOBI, AZW, EPUB, AZW3, FB2, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

Převedl jsem PDF soubor, ale výsledný soubor má různé problémy?

PDF is a terrible format to convert from. For a list of the various issues you will encounter when converting PDF, see: *Convert PDF documents* (stránka 68).

How do I convert my file containing non-English characters, or smart quotes?

There are two aspects to this problem:

1. Knowing the encoding of the source file: calibre tries to guess what character encoding your source files use, but often, this is impossible, so you need to tell it what encoding to use. This can be done in the GUI via the *Input character encoding* field in the *Look & Feel* section of the conversion dialog. The command-line tools all have an `--input-encoding` option.
2. When adding HTML files to calibre, you may need to tell calibre what encoding the files are in. To do this go to *Preferences->Plugins->File Type plugins* and customize the HTML2Zip plugin, telling it what encoding your HTML files are in. Now when you add HTML files to calibre they will be correctly processed. HTML files from different sources often have different encodings, so you may have to change this setting repeatedly. A common encoding for many files from the web is `cp1252` and I would suggest you try that first. Note that when converting HTML files, leave the input encoding setting mentioned above blank. This is because the HTML2ZIP plugin automatically converts the HTML files to a standard encoding (utf-8).

What's the deal with Table of Contents in MOBI files?

The first thing to realize is that most ebooks have two tables of contents. One is the traditional Table of Contents, like the ToC you find in paper books. This Table of Contents is part of the main document flow and can be styled however you like. This ToC is called the *content ToC*.

Then there is the *metadata ToC*. A metadata ToC is a ToC that is not part of the book text and is typically accessed by some special button on a reader. For example, in the calibre viewer, you use the Show Table of Contents button to see this ToC. This ToC cannot be styled by the book creator. How it is represented is up to the viewer program.

In the MOBI format, the situation is a little confused. This is because the MOBI format, alone amongst mainstream ebook formats, *does not* have decent support for a metadata ToC. A MOBI book simulates the presence of a metadata ToC by putting an *extra* content ToC at the end of the book. When you click Goto Table of Contents on your Kindle, it is to this extra content ToC that the Kindle takes you.

Now it might well seem to you that the MOBI book has two identical ToCs. Remember that one is semantically a content ToC and the other is a metadata ToC, even though both might have exactly the same entries and look the same. One can be accessed directly from the Kindle's menus, the other cannot.

When converting to MOBI, calibre detects the *metadata ToC* in the input document and generates an end-of-file ToC in the output MOBI file. You can turn this off by an option in the MOBI Output settings. You can also tell calibre whether to put it and the start or the end of the book via an option in the MOBI Output settings. Remember this ToC is semantically a *metadata ToC*, in any format other than MOBI it *cannot not be part of the text*. The fact that it is part of the text in MOBI is an accident caused by the limitations of MOBI. If you want a ToC at a particular location in your document text, create one by hand. So we strongly recommend that you leave the default as it is, i.e. with the metadata ToC at the end of the book. Also note that if you disable the generation of the end-of-file ToC the resulting MOBI file may not function correctly on a Kindle, since the Kindle's use the metadata ToC for many things, including the Page Flip feature.

If you have a hand edited ToC in the input document, you can use the ToC detection options in calibre to automatically generate the metadata ToC from it. See the conversion section of the User Manual for more details on how to use these options.

Finally, I encourage you to ditch the content ToC and only have a metadata ToC in your ebooks. Metadata ToCs will give the people reading your ebooks a much superior navigation experience (except on the Kindle, where they are essentially the same as a content ToC).

Poznámka: The newer AZW3 format has proper support for a metadata ToC. However, the Kindle firmware tends to malfunction if you disable the generation of the end-of-file inline ToC. So it is recommended that you leave the generated ToC alone. If you create an AZW3 file with a metadata ToC and no end-of-file generated ToC, some features on the Kindle will not work, such as the Page Flip feature.

The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.

This is caused by a bug in the Amazon software. You can work around it by going to *Preferences->Output Options->MOBI output* and setting the "Enable sharing of book content" option. If you are reconverting a previously converted book, you will also have to enable the option in the conversion dialog for that individual book (as per book conversion settings are saved and take precedence).

Note that doing this will mean that the generated MOBI will show up under personal documents instead of Books on the Kindle Fire and Amazon whispersync will not work, but the covers will. It's your choice which functionality is more important to you. I encourage you to contact Amazon and ask them to fix this bug.

The bug in Amazon's software is that when you put a MOBI file on a Kindle, unless the file is marked as a Personal document, Amazon assumes you bought the book from it and tries to download the cover thumbnail for it from its servers. When the download fails, it refuses to fallback to the cover defined in the MOBI file. This is likely deliberate on Amazon's part to try to force authors to sell only through them. In other words, Kindle's only display covers for books marked as Personal Documents or books bought directly from Amazon.

If you send a MOBI file to an e-ink Kindle with calibre using a USB connection, calibre works around this Amazon bug by uploading a cover thumbnail itself. However, that workaround is only possible when using a USB connection and sending with calibre. Note that if you send using email, Amazon will automatically mark the MOBI file as a Personal Document and the cover will work, but the book will show up in Personal Documents.

Jak převedu kolekci HTML souborů v určitém pořadí?

In order to convert a collection of HTML files in a specific order, you have to create a table of contents file. That is, another HTML file that contains links to all the other files in the desired order. Such a file looks like:

```
<html>
   <body>
     <h1>Table of Contents</h1>
     <p style="text-indent:0pt">
        <a href="file1.html">First File</a><br/>
        <a href="file2.html">Second File</a><br/>
        .
        .
        .
     </p>
   </body>
</html>
```

Then, just add this HTML file to the GUI and use the convert button to create your ebook. You can use the option in the Table of Contents section in the conversion dialog to control how the Table of Contents is generated.

Poznámka: By default, when adding HTML files, calibre follows links in the files in *depth first* order. This means that if file A.html links to B.html and C.html and D.html, but B.html also links to D.html, then the files will be in the order A.html, B.html, D.html, C.html. If instead you want the order to be A.html, B.html, C.html, D.html then you must tell calibre to add your files in *breadth first* order. Do this by going to Preferences->Plugins and customizing the HTML to ZIP plugin.

The EPUB I produced with calibre is not valid?

calibre does not guarantee that an EPUB produced by it is valid. The only guarantee it makes is that if you feed it valid XHTML 1.1 + CSS 2.1 it will output a valid EPUB. calibre tries hard to ensure that EPUBs it produces actually work as intended on a wide variety of devices, a goal that is incompatible with producing valid EPUBs, and one that is far more important to the vast majority of its users. If you need a tool that always produces valid EPUBs, calibre is not for you. This means, that if you want to send a calibre produced EPUB to an online store that uses an EPUB validity checker, you have to make sure that the EPUB is valid yourself, calibre will not do it for you – in other words you must feed calibre valid XHTML + CSS as the input documents.

How do I use some of the advanced features of the conversion tools?

You can get help on any individual feature of the converters by mousing over it in the GUI or running `ebook-convert dummy.html .epub -h` at a terminal. A good place to start is to look at the following demo file that demonstrates some of the advanced features html-demo.zip[36]

1.8.2 Device Integration

Obsah

- *What devices does calibre support?* (stránka 108)
- *How can I help get my device supported in calibre?* (stránka 109)
- *My device is not being detected by calibre?* (stránka 109)
- *My device is non-standard or unusual. What can I do to connect to it?* (stránka 109)
- *How does calibre manage collections on my SONY reader?* (stránka 110)
- *Can I use both calibre and the SONY software to manage my reader?* (stránka 110)
- *How do I use calibre with my iPad/iPhone/iPod touch?* (stránka 111)
- *How do I use calibre with my Android phone/tablet or Kindle Fire HD?* (stránka 112)
- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (stránka 113)
- *I get the error message "Failed to start content server: Port 8080 not free on '0.0.0.0''?* (stránka 113)
- *I cannot send emails using calibre?* (stránka 113)
- *Why is my device not detected in linux?* (stránka 114)
- *My device is getting mounted read-only in linux, so calibre cannot connect to it?* (stránka 114)
- *Why does calibre not support collections on the Kindle or shelves on the Nook?* (stránka 114)
- *I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?* (stránka 115)
- *I transferred some books to my Kindle using calibre and they did not show up?* (stránka 115)

What devices does calibre support?

calibre can directly connect to all the major (and most of the minor) ebook reading devices, smarthphones, tablets, etc. In addition, using the *Connect to folder* function you can use it with any ebook reader that exports itself as a USB disk.

[36] http://calibre-ebook.com/downloads/html-demo.zip

You can even connect to Apple devices (via iTunes), using the *Connect to iTunes* function.

How can I help get my device supported in calibre?

If your device appears as a USB disk to the operating system, adding support for it to calibre is very easy. We just need some information from you:

- Complete list of ebook formats that your device supports.
- Is there a special directory on the device in which all ebook files should be placed? Also does the device detect files placed in sub-directories?
- We also need information about your device that calibre will collect automatically. First, if your device supports SD cards, insert them. Then connect your device to the computer. In calibre go to *Preferences->Miscellaneous* and click the "Debug device detection" button. This will create some debug output. Copy it to a file and repeat the process, this time with your device disconnected from your computer.
- Send both the above outputs to us with the other information and we will write a device driver for your device.

Once you send us the output for a particular operating system, support for the device in that operating system will appear in the next release of calibre. To send us the output, open a bug report and attach the output to it. See calibre bugs[37].

My device is not being detected by calibre?

Následujte tyto kroky pro nalezení problému:

- Make sure that you are connecting only a single device to your computer at a time. Do not have another calibre supported device like an iPhone/iPad etc. at the same time.
- If you are connecting an Apple iDevice (iPad, iPod Touch, iPhone), use the 'Connect to iTunes' method in the 'Getting started' instructions in Calibre + Apple iDevices: Start here[38].
- Make sure you are running the latest version of calibre. The latest version can always be downloaded from the calibre website[39]. You can tell what version of calibre you are currently running by looking at the bottom line of the main calibre window.
- Ensure your operating system is seeing the device. That is, the device should show up in Windows Explorer (in Windows) or Finder (in OS X).
- In calibre, go to *Preferences->Ignored Devices* and check that your device is not being ignored
- If all the above steps fail, go to *Preferences->Miscellaneous* and click *Debug device detection* with your device attached and post the output as a ticket on the calibre bug tracker[40].

My device is non-standard or unusual. What can I do to connect to it?

In addition to the *Connect to Folder* function found under the Connect/Share button, calibre provides a `User Defined` device plugin that can be used to connect to any USB device that shows up as a disk drive in your operating system. Note: on Windows, the device must have a drive letter for calibre to use it. See the device plugin `Preferences -> Plugins -> Device Plugins -> User Defined` and `Preferences -> Miscellaneous -> Get information to setup the user defined device` for more information. Note that if you are using the user defined plugin for a device normally detected by a builtin calibre plugin, you must disable the builtin plugin first, so that your user defined plugin is used instead.

[37] http://calibre-ebook.com/bugs
[38] http://www.mobileread.com/forums/showthread.php?t=118559
[39] http://calibre-ebook.com/download
[40] https://bugs.launchpad.net/calibre

How does calibre manage collections on my SONY reader?

When calibre connects with the reader, it retrieves all collections for the books on the reader. The collections of which books are members are shown on the device view.

When you send a book to the reader, calibre will add the book to collections based on the metadata for that book. By default, collections are created from tags and series. You can control what metadata is used by going to *Preferences->Plugins->Device Interface plugins* and customizing the SONY device interface plugin. If you remove all values, calibre will not add the book to any collection.

Collection management is largely controlled by the 'Metadata management' option found at *Preferences->Import/Export->Sending books to devices*. If set to 'Manual' (the default), managing collections is left to the user; calibre will not delete already existing collections for a book on your reader when you resend the book to the reader, but calibre will add the book to collections if necessary. To ensure that the collections for a book are based only on current calibre metadata, first delete the books from the reader, then resend the books. You can edit collections directly on the device view by double-clicking or right-clicking in the collections column.

If 'Metadata management' is set to 'Only on send', then calibre will manage collections more aggressively. Collections will be built using calibre metadata exclusively. Sending a book to the reader will correct the collections for that book so its collections exactly match the book's metadata, adding and deleting collections as necessary. Editing collections on the device view is not permitted, because collections not in the metadata will be removed automatically.

If 'Metadata management' is set to 'Automatic management', then calibre will update metadata and collections both when the reader is connected and when books are sent. When calibre detects the reader and generates the list of books on the reader, it will send metadata from the library to the reader for all books on the reader that are in the library (On device is True), adding and removing books from collections as indicated by the metadata and device customization. When a book is sent, calibre corrects the metadata for that book, adding and deleting collections. Manual editing of metadata on the device view is not allowed. Note that this option specifies sending metadata, not books. The book files on the reader are not changed.

In summary, choose 'manual management' if you want to manage collections yourself. Collections for a book will never be removed by calibre, but can be removed by you by editing on the device view. Choose 'Only on send' if you want calibre to manage collections when you send a book, adding books to and removing books from collections as needed. Choose 'Automatic management' if you want calibre to keep collections up to date whenever the reader is connected.

If you use multiple installations of calibre to manage your reader, then option 'Automatic management' may not be what you want. Connecting the reader to one library will reset the metadata to what is in that library. Connecting to the other library will reset the metadata to what is in that other library. Metadata in books found in both libraries will be flopped back and forth.

Can I use both calibre and the SONY software to manage my reader?

Yes, you can use both, provided you do not run them at the same time. That is, you should use the following sequence: Connect reader->Use one of the programs->Disconnect reader. Reconnect reader->Use the other program->disconnect reader.

The underlying reason is that the Reader uses a single file to keep track of 'meta' information, such as collections, and this is written to by both calibre and the Sony software when either updates something on the Reader. The file will be saved when the Reader is (safely) disconnected, so using one or the other is safe if there's a disconnection between them, but if you're not the type to remember this, then the simple answer is to stick to one or the other for the transfer and just export/import from/to the other via the computers hard disk.

If you do need to reset your metadata due to problems caused by using both at the same time, then just delete the media.xml file on the Reader using your PC's file explorer and it will be recreated after disconnection.

With recent reader iterations, SONY, in all its wisdom has decided to try to force you to use their software. If you install it, it auto-launches whenever you connect the reader. If you don't want to uninstall it altogether, there are a

couple of tricks you can use. The simplest is to simply re-name the executable file that launches the library program. More detail in the forums[41].

How do I use calibre with my iPad/iPhone/iPod touch?

Over the air

The easiest way to browse your calibre collection on your Apple device (iPad/iPhone/iPod) is by using the calibre content server, which makes your collection available over the net. First perform the following steps in calibre

- Set the Preferred Output Format in calibre to EPUB (The output format can be set under *Preferences->Interface->Behavior*)
- Set the output profile to iPad (this will work for iPhone/iPods as well), under *Preferences->Conversion->Common Options->Page Setup*
- Convert the books you want to read on your iDevice to EPUB format by selecting them and clicking the Convert button.
- Turn on the Content Server by clicking the *Connect/Share* button and leave calibre running. You can also tell calibre to automatically start the content server via *Preferences->Sharing over the net*.

There are many apps for your iDevice that can connect to calibre. Here we describe using two of them, iBooks and Stanza.

Using Stanza You should be able to access your books on your iPhone by opening Stanza. Go to "Get Books" and then click the "Shared" tab. Under Shared you will see an entry "Books in calibre". If you don't, make sure your iPad/iPhone is connected using the WiFi network in your house, not 3G. If the calibre catalog is still not detected in Stanza, you can add it manually in Stanza. To do this, click the "Shared" tab, then click the "Edit" button and then click "Add book source" to add a new book source. In the Add Book Source screen enter whatever name you like and in the URL field, enter the following:

```
http://192.168.1.2:8080/
```

Replace `192.168.1.2` with the local IP address of the computer running calibre. If you have changed the port the calibre content server is running on, you will have to change `8080` as well to the new port. The local IP address is the IP address you computer is assigned on your home network. A quick Google search will tell you how to find out your local IP address. Now click "Save" and you are done.

If you get timeout errors while browsing the calibre catalog in Stanza, try increasing the connection timeout value in the stanza settings. Go to Info->Settings and increase the value of Download Timeout.

Pomocí iBooks Start the Safari browser and type in the IP address and port of the computer running the calibre server, like this:

```
http://192.168.1.2:8080/
```

Replace `192.168.1.2` with the local IP address of the computer running calibre. If you have changed the port the calibre content server is running on, you will have to change `8080` as well to the new port. The local IP address is the IP address you computer is assigned on your home network. A quick Google search will tell you how to find out your local IP address.

You will see a list of books in Safari, just click on the epub link for whichever book you want to read, Safari will then prompt you to open it with iBooks.

[41] http://www.mobileread.com/forums/showthread.php?t=65809

Pomocí USB kabelu a iTunes

Use the 'Connect to iTunes' method in the 'Getting started' instructions in Calibre + Apple iDevices: Start here[42].

This method only works on Windows Vista and higher, and OS X up to 10.8. Linux is not supported (iTunes is not available in linux) and OS X newer than 10.8 is not supported, as Apple removed the facility to use iTunes to manage books, replacing it with iBooks.

How do I use calibre with my Android phone/tablet or Kindle Fire HD?

There are two ways that you can connect your Android device to calibre. Using a USB cable – or wirelessly, over the air. The first step to using an Android device is installing an ebook reading application on it. There are many free and paid ebook reading applications for Android: Some examples (in no particular order): FBReader[43], Moon+[44], Mantano[45], Aldiko[46], Kindle[47].

Pomocí USB kabelu

Simply plug your device into the computer with a USB cable. calibre should automatically detect the device and then you can transfer books to it by clicking the Send to Device button. calibre does not have support for every single android device out there, so if your device is not automatically detected, follow the instructions at *How can I help get my device supported in calibre?* (stránka 109) to get your device supported in calibre.

Poznámka: With newer Android devices, the USB connection is not supported on Windows XP and OS X before Mavericks (10.9). If you are on Windows XP or old versions of OS X, you should use one of the wireless connection methods.

Over the air

The easiest way to transfer books wirelessly to your Android device is to use the Calibre Companion[48] Android app. This app is maintained by a core calibre developer and allows calibre to connect to your Android device wirelessly, just as though you plugged in the device with a USB cable. You can browse files on the device in calibre and use the *Send to device* button to transfer files to your device wirelessly.

calibre also has a builtin web server, the *Content Server*. You can browse your calibre collection on your Android device by using the calibre content server, which makes your collection available over the net. First perform the following steps in calibre

- Set the *Preferred Output Format* in calibre to EPUB for normal Android devices or MOBI for Kindles (The output format can be set under *Preferences->Interface->Behavior*)
- Convert the books you want to read on your device to EPUB/MOBI format by selecting them and clicking the Convert button.
- Turn on the Content Server in calibre's preferences and leave calibre running.

Nyní na vašem Android zařízení otevřete prohlížeč a jděte na

http://192.168.1.2:8080/

[42] http://www.mobileread.com/forums/showthread.php?t=118559
[43] https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en
[44] https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en
[45] https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en
[46] https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en
[47] https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps
[48] http://www.multipie.co.uk/calibre-companion/

Replace `192.168.1.2` with the local IP address of the computer running calibre. If your local network supports the use of computer names, you can replace the IP address with the network name of the computer. If you have changed the port the calibre content server is running on, you will have to change `8080` as well to the new port.

The local IP address is the IP address you computer is assigned on your home network. A quick Google search will tell you how to find out your local IP address. You can now browse your book collection and download books from calibre to your device to open with whatever ebook reading software you have on your android device.

Some reading programs support browsing the calibre library directly. For example, in Aldiko, click My Catalogs, then + to add a catalog, then give the catalog a title such as "calibre" and provide the URL listed above. You can now browse the calibre library and download directly into the reading software.

Can I access my calibre books using the web browser in my Kindle or other reading device?

calibre has a *Content Server* that exports the books in calibre as a web page. You can turn it on under *Preferences->Sharing over the net*. Then just point the web browser on your device to the computer running the Content Server and you will be able to browse your book collection. For example, if the computer running the server has IP address 63.45.128.5, in the browser, you would type:

```
http://63.45.128.5:8080
```

Some devices, like the Kindle (1/2/DX), do not allow you to access port 8080 (the default port on which the content server runs. In that case, change the port in the calibre Preferences to 80. (On some operating systems, you may not be able to run the server on a port number less than 1024 because of security settings. In this case the simplest solution is to adjust your router to forward requests on port 80 to port 8080).

I get the error message "Failed to start content server: Port 8080 not free on '0.0.0.0"'?

The most likely cause of this is your antivirus program. Try temporarily disabling it and see if it does the trick.

I cannot send emails using calibre?

Because of the large amount of spam in email, sending email can be tricky, as different mail servers use different strategies to block email. The most common problem is if you are sending email directly (without a mail relay) in calibre. Many servers (for example, Amazon) block email that does not come from a well known relay. The most robust way to setup email sending in calibre is to do the following:

- Create a free GMX account at GMX[49].
- Goto *Preferences->Sharing by Email* in calibre and click the *Use GMX* button and fill in the information asked for.
- calibre will then use GMX to send the mail.
- If you are sending to your Kindle, remember to update the email preferences on your Amazon Kindle page to allow email sent from your GMX email address. Also note that Amazon does not allow email delivery of AZW3 and new style (KF8) MOBI files.

Even after doing this, you may have problems. One common source of problems is that some poorly designed antivirus programs block calibre from opening a connection to send email. Try adding an exclusion for calibre in your antivirus program.

Poznámka: Microsoft/Google/Gmx can disable your account if you use it to send large amounts of email. So, when using Hotmail/Gmail to send mail calibre automatically restricts itself to sending one book every five minutes. If you

[49] http://www.gmx.com

don't mind risking your account being blocked you can reduce this wait interval by going to Preferences->Tweaks in calibre.

Poznámka: Google recently deliberately broke their email sending protocol (SMTP) support in an attempt to force everyone to use their web interface so they can show you more ads. They are trying to claim that SMTP is insecure, that is incorrect and simply an excuse. If you have trouble with gmail you will need to allow less secure apps as described here[50].

Poznámka: If you are concerned about giving calibre access to your email account, simply create a new free email account with GMX or Hotmail and use it only for calibre.

Why is my device not detected in linux?

calibre needs your linux kernel to have been setup correctly to detect devices. If your devices are not detected, perform the following tests:

```
grep SYSFS_DEPRECATED /boot/config-`uname -r`
```

You should see something like `CONFIG_SYSFS_DEPRECATED_V2 is not set`. Also,

```
grep CONFIG_SCSI_MULTI_LUN /boot/config-`uname -r`
```

must return `CONFIG_SCSI_MULTI_LUN=y`. If you don't see either, you have to recompile your kernel with the correct settings.

My device is getting mounted read-only in linux, so calibre cannot connect to it?

Linux kernels mount devices read-only when their filesystems have errors. You can repair the filesystem with:

```
sudo fsck.vfat -y /dev/sdc
```

Replace /dev/sdc with the path to the device node of your device. You can find the device node of your device, which will always be under /dev by examining the output of:

```
mount
```

Why does calibre not support collections on the Kindle or shelves on the Nook?

Neither the Kindle nor the Nook provide any way to manipulate collections over a USB connection. If you really care about using collections, I would urge you to sell your Kindle/Nook and get a Kobo. Only Kobo seems to understand that life is too short to be entering collections one by one on an e-ink screen :)

Note that in the case of the Kindle, there is a way to manipulate collections via USB, but it requires that the Kindle be rebooted *every time* it is disconnected from the computer, for the changes to the collections to be recognized. As such, it is unlikely that any calibre developers will ever feel motivated enough to support it. There is however, a calibre plugin that allows you to create collections on your Kindle from the calibre metadata. It is available from here[51].

Poznámka: Amazon have removed the ability to manipulate collections completely in their newer models, like the Kindle Touch and Kindle Fire, making even the above plugin useless, unless you root your Kindle and install custom firmware.

[50] https://support.google.com/accounts/answer/6010255
[51] http://www.mobileread.com/forums/showthread.php?t=244202

I am getting an error when I try to use calibre with my Kobo Touch/Glo/etc.?

The Kobo has very buggy firmware. Connecting to it has been known to fail at random. Certain combinations of motherboard, USB ports/cables/hubs can exacerbate this tendency to fail. If you are getting an error when connecting to your touch with calibre try the following, each of which has solved the problem for *some* calibre users.

- Connect the Kobo directly to your computer, not via USB Hub
- Try a different USB cable and a different USB port on your computer
- Try a different computer, in particular the Kobo does not work well with some Windows XP machines. If you are on Windows XP, try a computer with a newer version of Windows.
- Try upgrading the firmware on your Kobo Touch to the latest
- Try resetting the Kobo (sometimes this cures the problem for a little while, but then it re-appears, in which case you have to reset again and again)
- Try only putting one or two books onto the Kobo at a time and do not keep large collections on the Kobo

I transferred some books to my Kindle using calibre and they did not show up?

Books sent to the Kindle only show up on the Kindle after they have been *indexed* by the Kindle. This can take some time. If the book still does not show up after some time, then it is likely that the Kindle indexer crashed. Sometimes a particular book can cause the indexer to crash. Unfortunately, Amazon has not provided any way to deduce which book is causing a crash on the Kindle. Your only recourse is to either reset the Kindle, or delete all files from its memory using Windows Explorer (or whatever file manager you use) and then send the books to it again, one by one, until you discover the problem book. Once you have found the problem book, delete it off the Kindle and do a MOBI to MOBI or MOBI to AZW3 conversion in calibre and then send it back. This will most likely take care of the problem.

1.8.3 Library Management

Obsah

- *What formats does calibre read metadata from?* (stránka 115)
- *Where are the book files stored?* (stránka 115)
- *How does calibre manage author names and sorting?* (stránka 116)
- *Why doesn't calibre let me store books in my own directory structure?* (stránka 117)
- *Why doesn't calibre have a column for foo?* (stránka 118)
- *Can I have a column showing the formats or the ISBN?* (stránka 118)
- *How do I move my calibre library from one computer to another?* (stránka 118)
- *The list of books in calibre is blank!* (stránka 118)
- *I am getting errors with my calibre library on a networked drive/NAS?* (stránka 119)

What formats does calibre read metadata from?

calibre reads metadata from the following formats: CHM, LRF, PDF, LIT, RTF, OPF, MOBI, PRC, EPUB, FB2, IMP, RB, HTML. In addition it can write metadata to: LRF, RTF, OPF, EPUB, PDF, MOBI

Where are the book files stored?

When you first run calibre, it will ask you for a folder in which to store your books. Whenever you add a book to calibre, it will copy the book into that folder. Books in the folder are nicely arranged into sub-folders by Author and

Title. Note that the contents of this folder are automatically managed by calibre, **do not** add any files/folders manually to this folder, as they may be automatically deleted. If you want to add a file associated to a particular book, use the top right area of *Edit metadata* dialog to do so. Then, calibre will automatically put that file into the correct folder and move it around when the title/author changes.

Metadata about the books is stored in the file `metadata.db` at the top level of the library folder This file is is a sqlite database. When backing up your library make sure you copy the entire folder and all its sub-folders.

The library folder and all it's contents make up what is called a calibre library. You can have multiple such libraries. To manage the libraries, click the calibre icon on the toolbar. You can create new libraries, remove/rename existing ones and switch between libraries easily.

You can copy or move books between different libraries (once you have more than one library setup) by right clicking on a book and selecting the *Copy to library* action.

How does calibre manage author names and sorting?

Author names are complex, especially across cultures, see this note[52] for some of complexities. calibre has a very flexible strategy for managing author names. The first thing to understand is that books and authors are separate entities in calibre. A book can have more than one author, and an author can have more than one book. You can manage the authors of a book by the edit metadata dialog. You can manage individual authors by right clicking on the author in the Tag Browser on the left of the main calibre screen and selecting *Manage authors*. Using this dialog you can change the name of an author and also how that name is sorted. This will automatically change the name of the author in all the books of that author. When a book has multiple authors, separate their names using the & character.

Now coming to author name sorting:

- When a new author is added to calibre (this happens whenever a book by a new author is added), calibre automatically computes a sort string for both the book and the author.
- Authors in the Tag Browser are sorted by the sort value for the **authors**. Remember that this is different from the Author sort field for a book.
- By default, this sort algorithm assumes that the author name is in `First name Last name` format and generates a `Last name, First name` sort value.
- You can change this algorithm by going to Preferences->Tweaks and setting the *author_sort_copy_method* tweak.
- You can force calibre to recalculate the author sort values for every author by right clicking on any author and selecting *Manage authors*, then pushing the *Recalculate all author sort values* button. Do this after you have set the author_sort_copy_method tweak to what you want.
- You can force calibre to recalculate the author sort values for all books by using the bulk metadata edit dialog (select all books and click edit metadata, check the *Automatically set author sort* checkbox, then press OK.)
- When recalculating the author sort values for books, calibre uses the author sort values for each individual author. Therefore, ensure that the individual author sort values are correct before recalculating the books' author sort values.
- You can control whether the Tag Browser display authors using their names or their sort values by setting the *categories_use_field_for_author_name* tweak in Preferences->Tweaks

Note that you can set an individual author's sort value to whatever you want using *Manage authors*. This is useful when dealing with names that calibre will not get right, such as complex multi-part names like Miguel de Cervantes Saavedra or when dealing with Asian names like Sun Tzu.

With all this flexibility, it is possible to have calibre manage your author names however you like. For example, one common req

[52] http://www.w3.org/International/questions/qa-personal-names.en.php?changelang=en

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Change all author names to LN, FN using the Manage authors dialog.
- After you have changed all the authors, press the *Recalculate all author sort values* button.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

Poznámka:

When changing from FN LN to LN, FN, it is often the case that the values in author_sort are already in LN, FN format. If this is

- set the `author_sort_copy_method` tweak to `copy` as described above.
- restart calibre. Do not change any book metadata before doing the remaining steps.
- open the Manage authors dialog. Press the `copy all author sort values to author` button.
- Check through the authors to be sure you are happy. You can still press Cancel to abandon the changes. Once you press OK, there is no undo.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

Why doesn't calibre let me store books in my own directory structure?

The whole point of calibre's library management features is that they provide a search and sort based interface for locating books that is *much* more efficient than any possible directory scheme you could come up with for your collection. Indeed, once you become comfortable using calibre's interface to find, sort and browse your collection, you wont ever feel the need to hunt through the files on your disk to find a book again. By managing books in its own directory structure of Author -> Title -> Book files, calibre is able to achieve a high level of reliability and standardization. To illustrate why a search/tagging based interface is superior to folders, consider the following. Suppose your book collection is nicely sorted into folders with the following scheme:

```
Genre -> Author -> Series -> ReadStatus
```

Now this makes it very easy to find for example all science fiction books by Isaac Asimov in the Foundation series. But suppose you want to find all unread science fiction books. There's no easy way to do this with this folder scheme, you would instead need a folder scheme that looks like:

```
ReadStatus -> Genre -> Author -> Series
```

In calibre, you would instead use tags to mark genre and read status and then just use a simple search query like `tag:scifi and not tag:read`. calibre even has a nice graphical interface, so you don't need to learn its search language instead you can just click on tags to include or exclude them from the search.

To those of you that claim that you need access to the filesystem, so that you can have access to your books over the network, calibre has an excellent content server that gives you access to your calibre library over the net.

If you are worried that someday calibre will cease to be developed, leaving all your books marooned in its folder structure, explore the powerful *Save to Disk* feature in calibre that lets you export all your files into a folder structure of arbitrary complexity based on their metadata.

Finally, the reason there are numbers at the end of every title folder, is for *robustness*. That number is the id number of the book record in the calibre database. The presence of the number allows you to have multiple records with the same title and author names. It is also part of what allows calibre to magically regenerate the database with all metadata if the database file gets corrupted. Given that calibre's mission is to get you to stop storing metadata in filenames and stop using the filesystem to find things, the increased robustness afforded by the id numbers is well worth the uglier folder names.

If you are still not convinced, then I'm afraid calibre is not for you. Look elsewhere for your book cataloguing needs. Just so we're clear, **this is not going to change**. Kindly do not contact us in an attempt to get us to change this.

Why doesn't calibre have a column for foo?

calibre is designed to have columns for the most frequently and widely used fields. In addition, you can add any columns you like. Columns can be added via *Preferences->Interface->Add your own columns*. Watch the tutorial UI Power tips[53] to learn how to create your own columns, or read this blog post[54].

You can also create "virtual columns" that contain combinations of the metadata from other columns. In the add column dialog use the *Quick create* links to easily create columns to show the book ISBN or formats. You can use the powerful calibre template language to do much more with columns. For more details, see *The calibre template language* (stránka 133).

Can I have a column showing the formats or the ISBN?

Yes, you can. Follow the instructions in the answer above for adding custom columns.

How do I move my calibre library from one computer to another?

Simply copy the calibre library folder from the old to the new computer. You can find out what the library folder is by clicking the calibre icon in the toolbar. The very first item is the path to the library folder. Now on the new computer, start calibre for the first time. It will run the Welcome Wizard asking you for the location of the calibre library. Point it to the previously copied folder. If the computer you are transferring to already has a calibre installation, then the Welcome wizard wont run. In that case, right-click the calibre icon in the tooolbar and point it to the newly copied directory. You will now have two calibre libraries on your computer and you can switch between them by clicking the calibre icon on the toolbar. Transferring your library in this manner preserver all your metadata, tags, custom columns, etc.

Note that if you are transferring between different types of computers (for example Windows to OS X) then after doing the above you should also right-click the calibre icon on the tool bar, select Library Maintenance and run the Check Library action. It will warn you about any problems in your library, which you should fix by hand.

Poznámka: A calibre library is just a folder which contains all the book files and their metadata. All the metadata is stored in a single file called metadata.db, in the top level folder. If this file gets corrupted, you may see an empty list of books in calibre. In this case you can ask calibre to restore your books by doing a right-click on the calibre icon in the toolbar and selecting Library Maintenance->Restore database

---

The list of books in calibre is blank!

In order to understand why that happened, you have to understand what a calibre library is. At the most basic level, a calibre library is just a folder. Whenever you add a book to calibre, that book's files are copied into this folder (arranged into sub folders by author and title). Inside the calibre library folder, at the top level, you will see a file called metadata.db. This file is where calibre stores the metadata like title/author/rating/tags etc. for *every* book in your calibre library. The list of books that calibre displays is created by reading the contents of this metadata.db file.

There can be two reasons why calibre is showing a empty list of books:

[53] http://calibre-ebook.com/demo#tutorials
[54] http://blog.calibre-ebook.com/2011/11/calibre-custom-columns.html

- Your calibre library folder changed its location. This can happen if it was on an external disk and the drive letter for that disk changed. Or if you accidentally moved the folder. In this case, calibre cannot find its library and so starts up with an empty library instead. To remedy this, do a right-click on the calibre icon in the calibre toolbar and select Switch/create library. Click the little blue icon to select the new location of your calibre library and click OK. If you don't know the new location search your computer for the file `metadata.db`.
- Your metadata.db file was deleted/corrupted. In this case, you can ask calibre to rebuild the metadata.db from its backups. Right click the calibre icon in the calibre toolbar and select Library maintenance->Restore database. calibre will automatically rebuild metadata.db.

I am getting errors with my calibre library on a networked drive/NAS?

Do not put your calibre library on a networked drive.

A filesystem is a complex beast. Most network filesystems lack various filesystem features that calibre uses. Some don't support file locking, some don't support hardlinking, some are just flaky. Additionally, calibre is a single user application, if you accidentally run two copies of calibre on the same networked library, bad things will happen. Finally, different OSes impose different limitations on filesystems, so if you share your networked drive across OSes, once again, bad things *will happen*.

Consider using the calibre Content Server to make your books available on other computers. Run calibre on a single computer and access it via the Content Server or a Remote Desktop solution.

If you must share the actual library, use a file syncing tool like DropBox or rsync instead of a networked drive. If you are using a file-syncing tool it is **essential** that you make sure that both calibre and the file syncing tool do not try to access the calibre library at the same time. In other words, **do not** run the file syncing tool and calibre at the same time.

Even with these tools there is danger of data corruption/loss, so only do this if you are willing to live with that risk. In particular, be aware that **Google Drive** is incompatible with calibre, if you put your calibre library in Google Drive, **you will suffer data loss**. See this thread[55] for details.

1.8.4 Různé

[55] http://www.mobileread.com/forums/showthread.php?t=205581

Obsah

- *I want calibre to download news from my favorite news website.* (stránka 120)
- *Why the name calibre?* (stránka 120)
- *Why does calibre show only some of my fonts on OS X?* (stránka 120)
- *calibre is not starting on Windows?* (stránka 121)
- *calibre freezes/crashes occasionally?* (stránka 121)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (stránka 122)
- *calibre is not starting/crashing on OS X?* (stránka 122)
- *Stáhnul jsem instalační soubor, ale nefunguje?* (stránka 123)
- *My antivirus program claims calibre is a virus/trojan?* (stránka 123)
- *How do I backup calibre?* (stránka 123)
- *How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?* (stránka 124)
- *I am getting a "Permission Denied" error?* (stránka 124)
- *Can I have the comment metadata show up on my reader?* (stránka 124)
- *How do I get calibre to use my HTTP proxy?* (stránka 125)
- *I want some feature added to calibre. What can I do?* (stránka 125)
- *Why doesn't calibre have an automatic update?* (stránka 125)
- *How is calibre licensed?* (stránka 125)
- *How do I run calibre from my USB stick?* (stránka 126)
- *How do I run parts of calibre like news download and the content server on my own linux server?* (stránka 126)

I want calibre to download news from my favorite news website.

If you are reasonably proficient with computers, you can teach calibre to download news from any website of your choosing. To learn how to do this see *Přidávání oblíbených webových stránek se zprávami* (stránka 23).

Otherwise, you can request a particular news site by posting in the calibre Recipes forum[56].

Why the name calibre?

Take your pick:

- Convertor And LIBRary for Ebooks
- A high *calibre* product
- A tribute to the SONY Librie which was the first e-ink based ebook reader
- My wife chose it ;-)

calibre is pronounced as cal-i-ber *not* ca-li-bre. If you're wondering, calibre is the British/commonwealth spelling for caliber. Being Indian, that's the natural spelling for me.

Why does calibre show only some of my fonts on OS X?

calibre embeds fonts in ebook files it creates. Ebook files support embedding only TrueType and OpenType (.ttf and .otf) fonts. Most fonts on OS X systems are in .dfont format, thus they cannot be embedded. calibre shows only TrueType and OpenType fonts found on your system. You can obtain many such fonts on the web. Simply download the .ttf/.otf files and add them to the Library/Fonts directory in your home directory.

[56] http://www.mobileread.com/forums/forumdisplay.php?f=228

calibre is not starting on Windows?

There can be several causes for this:

- If you are on Windows XP, or on a computer with a processor that does not support SSE2 (such as AMD processors from before 2003) try installing calibre version 1.48[57]. calibre 2.0 and newer use Qt 5 which is known to be incompatible with Windows XP machines, and requires SSE2. Simply un-install calibre and then install version 1.48, doing so will not affect your books/settings.
- If you get an error about calibre not being able to open a file because it is in use by another program, do the following:
  - Uninstall calibre
  - Reboot your computer
  - Re-install calibre. But do not start calibre from the installation wizard.
  - Temporarily disable your antivirus program (disconnect from the Internet before doing so, to be safe)
  - Look inside the folder you chose for your calibre library. If you see a file named metadata.db, delete it.
  - Start calibre
  - From now on you should be able to start calibre normally.
- If you get an error about a Python function terminating unexpectedly after upgrading calibre, first uninstall calibre, then delete the folders (if they exists) `C:\Program Files\Calibre` and `C:\Program Files\Calibre2`. Now re-install and you should be fine.
- If you get an error in the welcome wizard on an initial run of calibre, try choosing a folder like `C:\library` as the calibre library (calibre sometimes has trouble with library locations if the path contains non-English characters, or only numbers, etc.)
- Try running it as Administrator (Right click on the icon and select "Run as Administrator")

If it still wont launch, start a command prompt (press the windows key and R; then type **`cmd.exe`** in the Run dialog that appears). At the command prompt type the following command and press Enter:

```
calibre-debug -g
```

Post any output you see in a help message on the Forum[58].

calibre freezes/crashes occasionally?

There are several possible things I know of, that can cause this:

- You recently connected an external monitor or TV to your computer. In this case, whenever calibre opens a new window like the edit metadata window or the conversion dialog, it appears on the second monitor where you don't notice it and so you think calibre has frozen. Disconnect your second monitor and restart calibre.
- If you use RoboForm, it is known to cause calibre to crash. Add calibre to the blacklist of programs inside RoboForm to fix this. Or uninstall RoboForm.
- The Logitech SetPoint Settings application causes random crashes in calibre when it is open. Close it before starting calibre.
- Constant Guard Protection by Xfinity causes crashes in calibre. You have to manually allow calibre in it or uninstall Constant Guard Protection.

[57] http://download.calibre-ebook.com/1.48.0/
[58] http://www.mobileread.com/forums/forumdisplay.php?f=166

- Spybot - Search & Destroy blocks calibre from accessing its temporary files breaking viewing and converting of books.
- You are using a Wacom branded USB mouse. There is an incompatibility between Wacom mice and the graphics toolkit calibre uses. Try using a non-Wacom mouse.
- On some 64 bit versions of Windows there are security software/settings that prevent 64-bit calibre from working properly. If you are using the 64-bit version of calibre try switching to the 32-bit version.
- If the crashes happen specifically when you are using a file open dialog, like clicking on the Add Books button or the Save to Disk button, then you may have an issue with the windows file open dialogs on your computer. Some calibre users have reported that uninstalling the SpiderOak encrypted backup software also fixes these crashes. If you do not wish to uninstall SpiderOak, you can also turn off "Enable OS integration" in the SpiderOak preferences.

If none of the above apply to you, then there is some other program on your computer that is interfering with calibre. First reboot your computer in safe mode, to have as few running programs as possible, and see if the crashes still happen. If they do not, then you know it is some program causing the problem. The most likely such culprit is a program that modifies other programs' behavior, such as an antivirus, a device driver, something like RoboForm (an automatic form filling app) or an assistive technology like Voice Control or a Screen Reader.

The only way to find the culprit is to eliminate the programs one by one and see which one is causing the issue. Basically, stop a program, run calibre, check for crashes. If they still happen, stop another program and repeat.

Using the viewer or doing any conversions results in a permission denied error on Windows

Something on your computer is preventing calibre from accessing its own temporary files. Most likely the permissions on your Temp folder are incorrect. Go to the folder file:*C:\Users\USERNAME\AppData\Local* in Windows Explorer and then right click on the file:*Temp* folder, select Properties and go to the Security tab. Make sure that your user account has full control for this folder.

Some users have reported that running the following command in an Administrator Command Prompt fixed their permissions. To get an Administrator Command Prompt search for cmd.exe in the start menu, then right click on the command prompt entry and select Run as Administrator. At the command prompt type the following command and press Enter:

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

Alternately, you can run calibre as Administrator, but doing so will cause some functionality, such as drag and drop to not work.

Finally, some users have reported that disabling UAC fixes the problem.

calibre is not starting/crashing on OS X?

One common cause of failures on OS X is the use of accessibility technologies that are incompatible with the graphics toolkit calibre uses. Try turning off VoiceOver if you have it on. Also go to System Preferences->System->Universal Access and turn off the setting for enabling access for assistive devices in all the tabs. Another cause can be some third party apps that modify system behavior, such as Smart Scroll.

You can obtain debug output about why calibre is not starting by running *Console.app*. Debug output will be printed to it. If the debug output contains a line that looks like:

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

then the problem is probably a corrupted font cache. You can clear the cache by following these instructions[59]. If that doesn't solve it, look for a corrupted font file on your system, in ~/Library/Fonts or the like. An easy way to check

[59] http://www.macworld.com/article/1139383/fontcacheclear.html

for corrupted fonts in OS X is to start the "Font Book" application, select all fonts and then in the File menu, choose "Validate fonts".

Stáhnul jsem instalační soubor, ale nefunguje?

Downloading from the Internet can sometimes result in a corrupted download. If the calibre installer you downloaded is not opening, try downloading it again. If re-downloading it does not work, download it from an alternate location[60]. If the installer still doesn't work, then something on your computer is preventing it from running.

- Try temporarily disabling your antivirus program (Microsoft Security Essentials, or Kaspersky or Norton or McAfee or whatever). This is most likely the culprit if the upgrade process is hanging in the middle.
- Try rebooting your computer and running a registry cleaner like Wise registry cleaner[61].
- Try a clean install. That is, uninstall calibre, delete `C:\Program Files\Calibre2` (or wherever you previously chose to install calibre). Then re-install calibre. Note that uninstalling does not touch your books or settings.
- Try downloading the installer with an alternate browser. For example if you are using Internet Explorer, try using Firefox or Chrome instead.
- If you get an error about a missing DLL on Windows, then most likely, the permissions on your temporary folder are incorrect. Go to the folder `C:\Users\USERNAME\AppData\Local` in Windows explorer and then right click on the `Temp` folder and select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

If you still cannot get the installer to work and you are on Windows, you can use the calibre portable install[62], which does not need an installer (it is just a zip file).

My antivirus program claims calibre is a virus/trojan?

The first thing to check is that you are downloading calibre from the official website: http://calibre-ebook.com/download. Make sure you are clicking the download links on the left, not the advertisements on the right. calibre is a very popular program and unscrupulous people try to setup websites offering it for download to fool the unwary.

If you have the official download and your antivirus program is still claiming calibre is a virus, then, your antivirus program is wrong. Antivirus programs use heuristics, patterns of code that "look suspicious" to detect viruses. It's rather like racial profiling. calibre is a completely open source product. You can actually browse the source code yourself (or hire someone to do it for you) to verify that it is not a virus. Please report the false identification to whatever company you buy your antivirus software from. If the antivirus program is preventing you from downloading/installing calibre, disable it temporarily, install calibre and then re-enable it.

How do I backup calibre?

The most important thing to backup is the calibre library folder, that contains all your books and metadata. This is the folder you chose for your calibre library when you ran calibre for the first time. You can get the path to the library folder by clicking the calibre icon on the main toolbar. You must backup this complete folder with all its files and sub-folders.

You can switch calibre to using a backed up library folder by simply clicking the calibre icon on the toolbar and choosing your backup library folder. A backed up library folder backs up your custom columns and saved searches as well as all your books and metadata.

[60] http://www.fosshub.com/Calibre.html/
[61] http://www.wisecleaner.com
[62] http://calibre-ebook.com/download_portable

If you want to backup the calibre configuration/plugins, you have to backup the config directory. You can find this config directory via *Preferences->Miscellaneous*. Note that restoring configuration directories is not officially supported, but should work in most cases. Just copy the contents of the backup directory into the current configuration directory to restore.

How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?

Most purchased EPUB books have DRM[63]. This prevents calibre from opening them. You can still use calibre to store and transfer them to your ebook reader. First, you must authorize your reader on a windows machine with Adobe Digital Editions. Once this is done, EPUB books transferred with calibre will work fine on your reader. When you purchase an epub book from a website, you will get an ".acsm" file. This file should be opened with Adobe Digital Editions, which will then download the actual ".epub" ebook. The ebook file will be stored in the folder "My Digital Editions", from where you can add it to calibre.

I am getting a "Permission Denied" error?

A permission denied error can occur because of many possible reasons, none of them having anything to do with calibre.

- You can get permission denied errors if you are using an SD card with write protect enabled.
- If you, or some program you used changed the file permissions of the files in question to read only.
- If there is a filesystem error on the device which caused your operating system to mount the filesystem in read only mode or mark a particular file as read only pending recovery.
- If the files have their owner set to a user other than you.
- If your file is open in another program.
- If the file resides on a device, you may have reached the limit of a maximum of 256 files in the root of the device. In this case you need to reformat the device/sd card referred to in the error message with a FAT32 filesystem, or delete some files from the SD card/device memory.

You will need to fix the underlying cause of the permissions error before resuming to use calibre. Read the error message carefully, see what file it points to and fix the permissions on that file or its containing folders.

Can I have the comment metadata show up on my reader?

Most readers do not support this. You should complain to the manufacturer about it and hopefully if enough people complain, things will change. In the meantime, you can insert the metadata, including comments into a "Jacket page" at the start of the ebook, by using the option to "Insert metadata as page at start of book" during conversion. The option is found in the *Structure Detection* section of the conversion settings. Note that for this to have effect you have to *convert* the book. If your book is already in a format that does not need conversion, you can convert from that format to the same format.

Another alternative is to create a catalog in ebook form containing a listing of all the books in your calibre library, with their metadata. Click-and-hold the convert button to access the catalog creation tool. And before you ask, no you cannot have the catalog "link directly to" books on your reader.

[63] http://drmfree.calibre-ebook.com/about#drm

How do I get calibre to use my HTTP proxy?

By default, calibre uses whatever proxy settings are set in your OS. Sometimes these are incorrect, for example, on Windows if you don't use Internet Explorer then the proxy settings may not be up to date. You can tell calibre to use a particular proxy server by setting the `http_proxy` environment variable. The format of the variable is: `http://username:password@servername` you should ask your network administrator to give you the correct value for this variable. Note that calibre only supports HTTP proxies not SOCKS proxies. You can see the current proxies used by calibre in Preferences->Miscellaneous.

I want some feature added to calibre. What can I do?

You have two choices:

1. Create a patch by hacking on calibre and send it to me for review and inclusion. See Development[64].
2. Open a bug requesting the feature[65] . Remember that while you may think your feature request is extremely important/essential, calibre developers might not agree. Fortunately, calibre is open source, which means you always have the option of implementing your feature yourself, or hiring someone to do it for you. Furthermore, calibre has a comprehensive plugin architecture, so you might be able to develop your feature as a plugin, see *Psaní vlastních modulů pro rozšíření funkcí Calibre* (stránka 170).

Why doesn't calibre have an automatic update?

Z mnoha důvodů:

- *There is no need to update every week*. If you are happy with how calibre works turn off the update notification and be on your merry way. Check back to see if you want to update once a year or so. There is a check box to turn off the update notification, on the update notification itself.
- calibre downloads currently use about 100TB of bandwidth a month[66]. Implementing automatic updates would greatly increase that and end up costing thousands of dollars a month, which someone has to pay.
- If I implement a dialog that downloads the update and launches it, instead of going to the website as it does now, that would save the most ardent calibre updater, *at most five clicks a week*. There are far higher priority things to do in calibre development.
- If you really, really hate downloading calibre every week but still want to be up to the latest, I encourage you to run from source, which makes updating trivial. Instructions are *available here* (stránka 260).
- There are third party automatic updaters for calibre made by calibre users in the calibre forum[67].

How is calibre licensed?

calibre is licensed under the GNU General Public License v3 (an open source license). This means that you are free to redistribute calibre as long as you make the source code available. So if you want to put calibre on a CD with your product, you must also put the calibre source code on the CD. The source code is available for download[68]. You are free to use the results of conversions from calibre however you want. You cannot use either code or libraries from calibre in your software without making your software open source. For details, see The GNU GPL v3[69].

[64] http://calibre-ebook.com/get-involved
[65] http://calibre-ebook.com/bugs
[66] http://calibre-ebook.com/dynamic/downloads
[67] http://www.mobileread.com/forums/forumdisplay.php?f=238
[68] http://download.calibre-ebook.com
[69] http://www.gnu.org/licenses/gpl.html

How do I run calibre from my USB stick?

A portable version of calibre is available here[70].

How do I run parts of calibre like news download and the content server on my own linux server?

First, you must install calibre onto your linux server. If your server is using a modern linux distro, you should have no problems installing calibre onto it.

Poznámka: calibre needs GLIBC >= 2.13 and libstdc++ >= 6.0.17. If you have an older server, you will either need to compile these from source, or use calibre 1.48 which requires only GLIBC >= 2.10. In addition, although the calibre command line utilities do not need a running X server, some of them do require the X server libraries to be installed on your system. This is because the use Qt, which links against these libraries. If you get an ImportError about some Qt modules, you are likely missing some X libraries.

You can run the calibre server via the command:

```
/opt/calibre/calibre-server --with-library /path/to/the/library/you/want/to/share
```

You can download news and convert it into an ebook with the command:

```
/opt/calibre/ebook-convert "Title of news source.recipe" outputfile.epub
```

If you want to generate MOBI, use outputfile.mobi instead and use `--output-profile kindle`.

You can email downloaded news with the command:

```
/opt/calibre/calibre-smtp
```

I leave figuring out the exact command line as an exercise for the reader.

Finally, you can add downloaded news to the calibre library with:

```
/opt/calibre/calibredb add --with-library /path/to/library outfile.epub
```

Remember to read the command line documentation section of the calibre User Manual to learn more about these, and other commands.

1.9 Kurzy

Zde naleznete kurzy, abyste mohli začít používat pokročilejší funkce Calibre, jako jsou XPath a šablony.

1.9.1 Správa podskupin knih, například "žánr"

Někteří lidé si chtějí uspořádat knihy ve své knihovně do podskupin podobných podsložkám. Nejběžněji udávaný důvod je vytvořit hierarchie žánrů, ale je mnoho dalších. Jeden uživatel požadoval způsob, jak uspořádat učebnice podle předmětu a čísla kurzu. Další uživatel chtěl sledovat dárky podle předmětu a příjemce. Tato výuka použije příklad žánru po zbytek tohoto příspěvku.

Než budete pokračovat, pamatujte, že nemluvíme o složkách na pevném disku. Podskupiny nejsou složky souborů. Knihy nebudou nikam kopírovány. Struktura souborů knihovny Calibre není ovlivněna. Namísto toho představujeme způsob, jak uspořádat a zobrazit podskupiny knih v knihovně Calibre.

[70] http://calibre-ebook.com/download_portable

- *Nastavení* (stránka 128)
- *Hledání* (stránka 130)
- *Omezení* (stránka 130)
- *Užitečné funkce šablon* (stránka 131)

Běžně udávané požadavky pro podskupiny, jako žánry, jsou:

- Podskupina (např. žánr) musí obsahovat (ukazovat na) knihy, ne kategorie knih. To odlišuje podskupiny od uživatelských kategorií Calibre.
- Kniha může být ve více podskupinách (žánrech). To odlišuje podskupiny od fyzických složek souborů.
- Podskupiny (žánry) musí tvořit hierarchii; podskupiny mohou obsahovat podskupiny.

Štítky vám poskytují první dva. Když označíte knihu štítkem žánru, můžete pak použít prohlížeč štítků (nebo hledat), abyste našli knihy s tímto žánrem, což je první. Mnoho knih může mít stejný štítek (štítky), to je druhý. Problém je, že štítky nevyhovují třetímu požadavku. Neposkytují hierarchii.

Funkce hierarchie Calibre vám poskytuje třetí, schopnost vidět žánry ve ‘stromu’ a schopnost snadno hledat knihy v žánry nebo podžánru. Například předpokládejme, že vaše struktura žánrů je podobná následující:

```
Genre
    . History
    .. Japanese
    .. Military
    .. Roman
    . Mysteries
    .. English
    .. Vampire
    . Science Fiction
    .. Alternate History
    .. Military
    .. Space Opera
```

```
. Thrillers
.. Crime
.. Horror
etc.
```

Použitím funkce hierarchie můžete vidět tyto žánry v prohlížeči štítků v podobě stromu, jak je zobrazeno na obrázku. V tomto příkladu je nejkrajnější úroveň (Žánr) vlastním sloupcem, který obsahuje žánry. Žánry obsahující podžánry se zobrazují s malým trojúhelníkem vedle sebe. Kliknutí na tento trojúhelník otevře položku a zobrazí podžánry, jak můžete vidět u Historie a Sci-fi.

Kliknutí na žánr může vyhledat všechny knihy s tímto žánrem nebo podřízené položky tohoto žánru. Například kliknutí na Sci-fi může předat všechny tři podřízené žánry, Alternativní historie, Vojenská, Vesmírná opera. Kliknutí na Alternativní historie předá knihy v tomto žánru, ignoruje ty v žánrech Vojenská a Vesmírná opera. Knihy samozřejmě mohou mít více žánrů. Pokud mát kniha žánr jak Vesmírná opera, tak Vojenská, pak uvidíte tuto knihu při kliknutí na oba žánry. Hledání je probráno podrobněji níže.

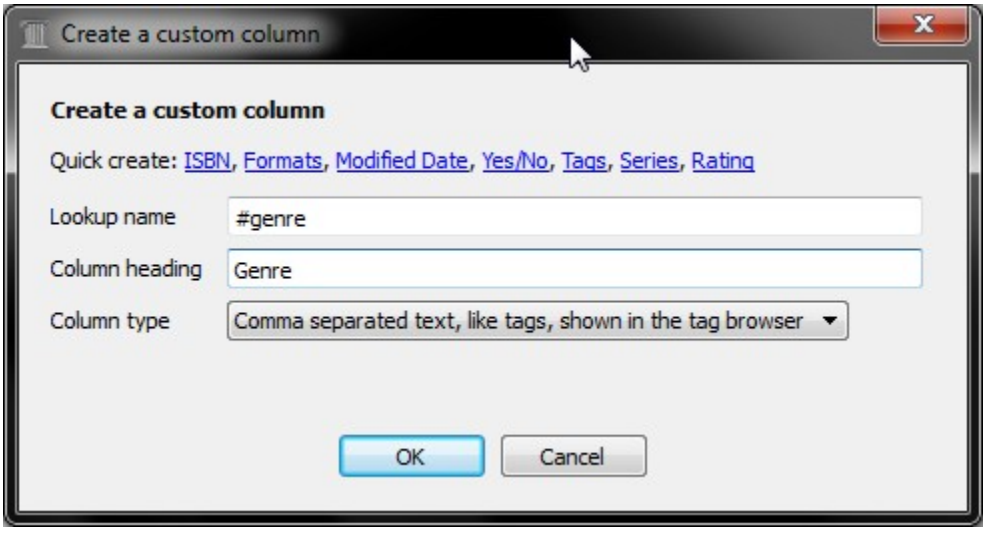
Další věc, kterou můžete vidět na obrázku, je, že žánry Vojenská se objevuje dvakrát, jednou pod Historie a jednou pod Sci-fi. Protože jsou žánry v hierarchii, jsou to dva samostatné žánry. Kniha může být v jednom, v druhém nebo (s pochybami v tomto případě) v obou. Například knihy Winstona Churchilla ze série "Druhá světová válka" by mohly být v "Historie.Vojenská". Knihy Davida Weber o Honor Harringtonové by mohly být v "Sci-fi.Vojenská" a ostatně také v "Sci-fi.Vesmírná opera".

Jakmile žánr existuje, tj. alespoň jedna kniha má tento žánr, můžete ho snadno použít na další knihy přetažením knih ze zobrazení knihovny na žánr, který chcete, aby knihy měly. Žánry můžete také použít v editorech metadat; více o tomto níže.

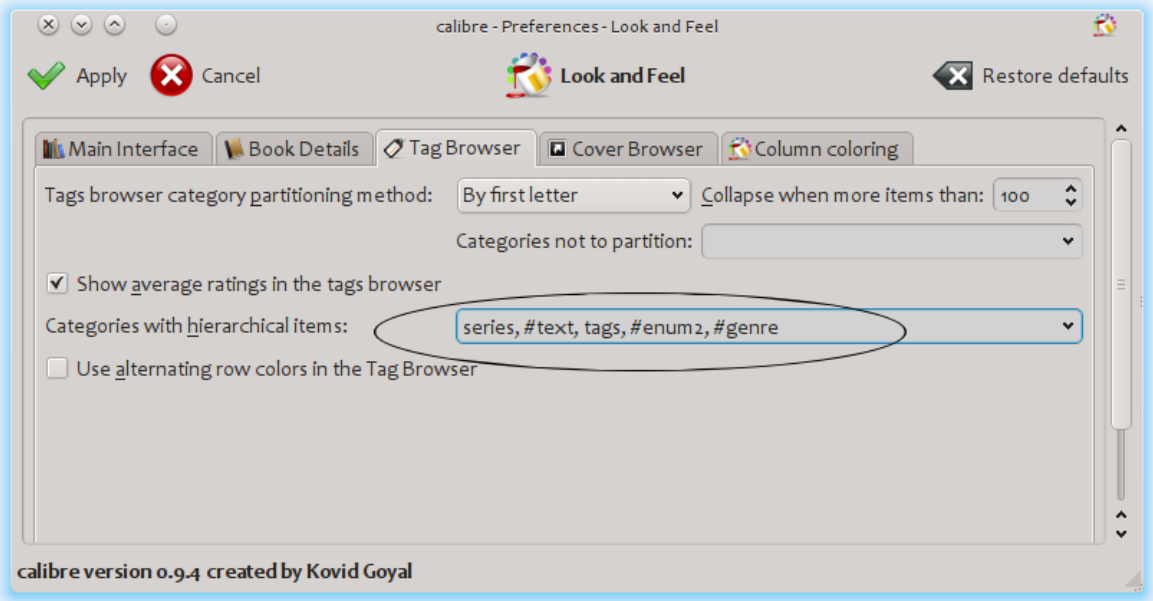
Nastavení

Teď může vaše otázka znít "Jak se tohle všechno děje? Jsou tři kroky: 1) vytvořte vlastní sloupec, 2) řekněte Calibre, že tento nový sloupec má být považován za hierarchii a 3) přidejte žánry.

Vlastní sloupec vytvoříte obvyklým způsobem pomocí Předvolby -> Přidat své vlastní sloupce. Tento příklad používá "#genre" jako název vyhledávání a "Žánr" jako záhlaví sloupce. Typ sloupce je "Čárkami oddělený text, jako jsou štítky, zobrazený v prohlížeči štítků".

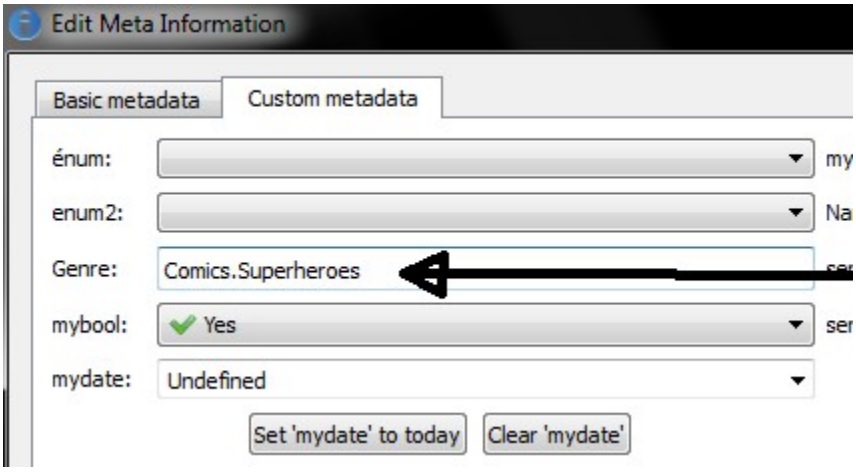


Pak po restartování Calibre musíte říct Calibre, že sloupec má být považován za hierarchii. Přejděte do Předvolby -> Vzhled -> Prohlížeč štítků a zadejte název vyhledávání "#zanr" do pole "Kategorie s hierarchickými položkami". Stiskněte Použít a jste hotovi s nastavením.



V tuto chvíli nejsou v tomto sloupci žádné žánry. Zbývá nám poslední krok: jak použít žánr na knihu. Žánr v Calibre neexistuje, dokud se neobjeví alespoň u jedné knihy. Pro naučení se, jak poprvé použít žánr, musíme přejít na podrobnosti o tom, jak žánr vypadá v metadatech knihy.

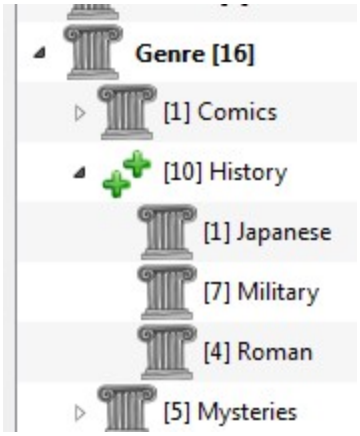
Hierarchie 'věcí' je vystavěna vytvořením položky skládající se z frází oddělených tečkami. V pokračování příkladu se žánrem by tyto položky byly "Historie.Vojenská", "Mysteriózní.Upíři", "Sci-fi.Vesmírná opera" atd. Tudíž pro vytvoření nového žánru vyberete knihu, která by měla mít tento žánr, upravte její metadata a zadejte nový žánr do sloupce, který jste vytvořili. V pokračování našeho příkladu, pokud chcete ke knize přiřadit nový žánr "Komiks" s podžánrem "Superhrdinové", otevřeli byste Upravit metadata pro tuto knihu (komiks) kniha, zvolili kartu Vlastní metadata a pak zadali "Komiks.Superhrdinové", jak je zobrazeno na následujícím obrázku (ignorujte ostatní vlastní sloupce):



Po provedení výše uvedeného uvidíte v prohlížeči štítků:

Od teď pro použití tohoto nového žánru na knihu (pravděpodobně komiksu) můžete buď přetáhnout knihu na žánru, nebo ho přidat do knihy pomocí úpravy metadat přesně stejným způsobem, jak je provedeno výše.

Hledání



Nejsnadnější způsob hledání žánrů je pomocí prohlížeče štítků, kliknutím na žánr, který chcete vidět. Kliknutím na žánr s podřízenými položkami vám zobrazí knihy s tímto žánrem a se všemi podřízenými žánry. To však může vyvolat otázku. Jen proto, že má žánr podřízené položky, neznamená to, že to není žánr sám o sobě. Například kniha může mít žánr “Historie”, ale ne “Historie.Vojenská”. Jak hledat knihy, které mají pouze “Historie”?

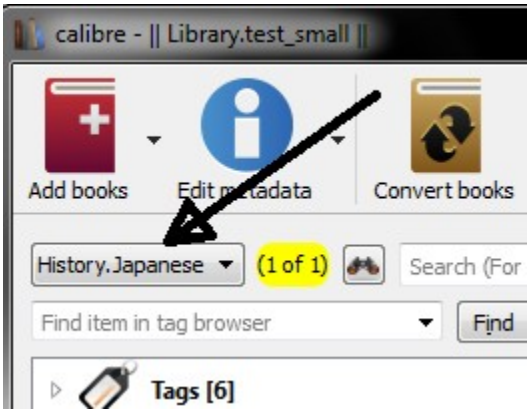
Mechanismus hledání prohlížeče štítku ví, pokud má položka podřízené položky. Pokud ano, klepnutí na položku proběhne 5 hledáními namísto normálních tří. První je normální zelené plus, které vám zobrazí knihy pouze s tímto žánrem(např. Historie). Druhé je dvojité plus (zobrazeno výše), která vám ukáže knihy s tímto žánrem a všemi podžánry (např. Historie a Historie.Vojenská). Třetí je normální červené minus, které vám ukáže knihy bez tohoto přesného žánru. Čtvrté je dvojité minus, které vám ukáže knihy, bez tohoto žánru nebo podžánrů. Páté je zpět na začátek, bez označení, což znamená žádné hledání.

Omezení

Pokud hledáte žánr, pak k tomu vytvořte uložené hledání, můžete použít pole ‘omezit na’ pro vytvoření virtuální knihovny knih s tímto žánrem. To je užitečné, pokud chcete provést další hledání v rámci žánru, nebo spravovat nebo aktualizovat metadata knih v žánru. V pokračování našeho příkladu můžete vytvořit uložené hledání s názvem “Historie.Japonská” nejdříve kliknutím na žánr Japonská v prohlížeči štítků pro získání hledání do pole hledání, zadáním

Historie.Japonská do pole uloženého hledání, a pak stisknutím tlačítka “uložit hledání” (zelené pole s bílým plus na pravé straně).

Po vytvoření uloženého hledání ho můžete použít jako omezení.



Užitečné funkce šablon

Možná budete chtít použít informace o žánru v šabloně, jako u uložení na disk nebo odeslat do zařízení. Otázkou by pak mohlo být “Jak získám nejkrajnější název nebo názvy žánru?” K dispozici je funkce šablony Calibre, podpoložky, aby to udělala snadněji.

Předpokládejme například, že chcete přidat nejkrajnější úroveň žánru do šablony uložit na disk pro vytvoření složek žánrů, jako v “Historie/Blížící se bouře - Churchill, Winston”. Abyste to provedli, musíte extrahovat první úroveň hierarchie a přidat ji před spolu s lomítkem pro označení, že by se měla vytvořit složka. Níže uvedená šablona toto splňuje:

```
{#genre:subitems(0,1)||/}{title} - {authors}
```

Pro více informací o šablonách a funkci `subitems()` se podívejte na *Jazyk šablony* (stránka 133).

1.9.2 Kurz XPath

V tomto kurzu vám bude poskytnut mírný úvod do XPath[71], dotazovací jazyk, který může být použit pro výběr libovolných částí dokumentů HTML[72] v calibre. XPath je široce používaný standard a jeho vyhledání na Google vám poskytne mnoho informací. Tento kurz se však zaměřuje na používání XPath pro úkoly spojené s e-knihami, jako je hledání nadpisů kapitol v nestrukturovaném dokumentu HTML.

[71] http://en.wikipedia.org/wiki/XPath
[72] http://en.wikipedia.org/wiki/HTML

Obsah

- *Výběr podle názvu značky* (stránka 132)
- *Výběr podle atributů* (stránka 132)
- *Výběr podle obsahu značky* (stránka 133)
- *Ukázková e-kniha* (stránka 133)
- *Vestavěné funkce XPath* (stránka 133)

Výběr podle názvu značky

Nejjednodušší forma výběru je vybrat značky podle názvu. Například předpokládejme, že chcete vybrat v dokumentu všechny značky `<h2>`. Dotaz XPath pro toto je jednoduše:

```
//h:h2        (Selects all <h2> tags)
```

Předpona *//* znamená *hledat na jakékoliv úrovni dokumentu*. Nyní předpokládejme, že chcete hledat značky `<span>`, které jsou uvnitř značek `<a>`. Toho může být dosaženo pomocí:

```
//h:a/h:span    (Selects <span> tags inside <a> tags)
```

Pokud chcete hledat značky na určité úrovni v dokumentu, změňte předponu:

```
/h:body/h:div/h:p (Selects <p> tags that are children of <div> tags that are
            children of the <body> tag)
```

Toto porovná pouze `<p>Velice krátká e-kniha pro ukázání použití XPath.</p>` v *Ukázková e-kniha* (stránka 133), ale ne žádné další značky `<p>`. Předpona `h:` ve výše uvedených příkladech je zapotřebí pro porovnání značek XHTML. To je proto, že calibre vnitřně představuje celý obsah jako XHTML. V XHTML mají značky *obor názvů* a `h:` je předpona oboru názvů pro značky HTML.

Nyní předpokládejme, že chcete vybrat jak značky `<h1>`, tak `<h2>`. K tomu potřebujeme konstrukci XPath zvanou *predikát*. *Predikát* je jednoduše test, který je použitý pro výběr značek. Testy mohou být libovolně výkonné, a jak bude tento kurz postupovat, uvidíte výkonnější příklady. Predikát je vytvořen uzavřením testovacího výrazu do hranatých závorek:

```
//*[name()='h1' or name()='h2']
```

V tomto výrazu XPath je několik nových funkcí. První je použití zástupného znaku `*`. Ten znamená *porovnat jakoukoliv značku*. Nyní se podívejte na testovací výraz `name()='h1' or name()='h2'`. *name()* je příklad *vestavěné funkce*. Jednoduše vyhodnocuje podle názvu značky. Takže jejím použitím můžeme vybrat značky, jejichž názvy jsou buď *h1* nebo *h2*. Pamatujte, že funkce *name()* ignoruje obory názvů, takže není potřeba předpona `h:`. XPath má několik užitečných vestavěných funkcí. V tomto kurzu bude představeno několik dalších.

Výběr podle atributů

Pro výběr značek na základě jejich atributů je požadováno použití predikátů:

```
//*[@style]              (Select all tags that have a style attribute)
//*[@class="chapter"]    (Select all tags that have class="chapter")
//h:h1[@class="bookTitle"] (Select all h1 tags that have class="bookTitle")
```

Operátor `@` zde odkazuje na atributy značky. Můžete použít některé z *vestavěných funkcí XPath* (stránka 133) pro provedení propracovanějšího porovnání hodnot atributů.

Výběr podle obsahu značky

Pomocí XPath můžete dokonce vybrat značky na základě textu, který obsahují. Nejlepší způsob, jak to udělat, je použít sílu *regulárních výrazů* pomocí vestavěné funkce *re:test()*:

```
//h:h2[re:test(., 'chapter|section', 'i')] (Selects <h2> tags that contain the words chapter or
                                          section)
```

Operátor `.` zde odkazuje na obsah značky, stejně jako operátor `@` odkazoval na její atributy.

Ukázková e-kniha

```
<html>
    <head>
        <title>A very short ebook</title>
        <meta name="charset" value="utf-8" />
    </head>
    <body>
        <h1 class="bookTitle">A very short ebook</h1>
        <p style="text-align:right">Written by Kovid Goyal</p>
        <div class="introduction">
            <p>A very short ebook to demonstrate the use of XPath.</p>
        </div>

        <h2 class="chapter">Chapter One</h2>
        <p>This is a truly fascinating chapter.</p>

        <h2 class="chapter">Chapter Two</h2>
        <p>A worthy continuation of a fine tradition.</p>
    </body>
</html>
```

Vestavěné funkce XPath

name() Název aktuální značky.

contains() `contains(s1, s2)` vrací *true*, pokud s1 obsahuje s2.

re:test() `re:test(zdroj, vzor, příznaky)` vrací *true*, pokud řetězec *zdroj* odpovídá regulárnímu výrazu *vzor*. Obzvlášť užitečný příznak je `i`, porovnání je pak bez rozlišení malých a velkých písmen. Dobré základy syntaxe pro regulární výrazy můžete nalézt na syntaxe regexp[73]

1.9.3 The calibre template language

The calibre template language is used in various places. It is used to control the folder structure and file name when saving files from the calibre library to the disk or eBook reader. It is also used to define "virtual" columns that contain data from other columns and so on.

The basic template language is very simple, but has very powerful advanced features. The basic idea is that a template consists of text and names in curly brackets that are then replaced by the corresponding metadata from the book being processed. So, for example, the default template used for saving books to device in calibre is:

```
{author_sort}/{title}/{title} - {authors}
```

[73] https://docs.python.org/2.7/library/re.html

For the book “The Foundation” by “Isaac Asimov” it will become:

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

The slashes are text, which is put into the template where it appears. For example, if your template is:

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

For the book “The Foundation” by “Isaac Asimov” it will become:

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

You can use all the various metadata fields available in calibre in a template, including any custom columns you have created yourself. To find out the template name for a column simply hover your mouse over the column header. Names for custom fields (columns you have created yourself) always have a # as the first character. For series type custom fields, there is always an additional field named `#seriesname_index` that becomes the series index for that series. So if you have a custom series field named `#myseries`, there will also be a field named `#myseries_index`.

In addition to the column based fields, you also can use:

```
{formats} - A list of formats available in the calibre library for a book
{identifiers:select(isbn)} - The ISBN number of the book
```

If a particular book does not have a particular piece of metadata, the field in the template is automatically removed for that book. Consider, for example:

```
{author_sort}/{series}/{title} {series_index}
```

If a book has a series, the template will produce:

```
Asimov, Isaac/Foundation/Second Foundation 3
```

and if a book does not have a series:

```
Asimov, Isaac/Second Foundation
```

(calibre automatically removes multiple slashes and leading or trailing spaces).

Advanced formatting

You can do more than just simple substitution with the templates. You can also conditionally include text and control how the substituted data is formatted.

First, conditionally including text. There are cases where you might want to have text appear in the output only if a field is not empty. A common case is `series` and `series_index`, where you want either nothing or the two values with a hyphen between them. Calibre handles this case using a special field syntax.

For example, assume you want to use the template:

```
{series} - {series_index} - {title}
```

If the book has no series, the answer will be `- - title`. Many people would rather the result be simply `title`, without the hyphens. To do this, use the extended syntax `{field:|prefix_text|suffix_text}`. When you use this syntax, if field has the value SERIES then the result will be `prefix_textSERIESsuffix_text`. If field has no value, then the result will be the empty string (nothing); the prefix and suffix are ignored. The prefix and suffix can contain blanks. **Do not use subtemplates (‘{ ... }‘) or functions (see below) as the prefix or the suffix.**

Using this syntax, we can solve the above series problem with the template:

```
{series}{series_index:| - | - }{title}
```

The hyphens will be included only if the book has a series index, which it will have only if it has a series.

Notes: you must include the : character if you want to use a prefix or a suffix. You must either use no | characters or both of them; using one, as in `{field:| - }`, is not allowed. It is OK not to provide any text for one side or the other, such as in `{series:|| - }`. Using `{title:||}` is the same as using `{title}`.

Second: formatting. Suppose you wanted to ensure that the series_index is always formatted as three digits with leading zeros. This would do the trick:

```
{series_index:0>3s} - Three digits with leading zeros
```

If instead of leading zeros you want leading spaces, use:

```
{series_index:>3s} - Three digits with leading spaces
```

For trailing zeros, use:

```
{series_index:0<3s} - Three digits with trailing zeros
```

If you use series indices with sub values (e.g., 1.1), you might want to ensure that the decimal points line up. For example, you might want the indices 1 and 2.5 to appear as 01.00 and 02.50 so that they will sort correctly. To do this, use:

```
{series_index:0>5.2f} - Five characters, consisting of two digits with leading zeros, a decimal point
```

If you want only the first two letters of the data, use:

```
{author_sort:.2} - Only the first two letter of the author sort name
```

The calibre template language comes from python and for more details on the syntax of these advanced formatting operations, look at the Python documentation[74].

Advanced features

Using templates in custom columns

There are sometimes cases where you want to display metadata that calibre does not normally display, or to display data in a way different from how calibre normally does. For example, you might want to display the ISBN, a field that calibre does not display. You can use custom columns for this by creating a column with the type 'column built from other columns' (hereafter called composite columns), and entering a template. Result: calibre will display a column showing the result of evaluating that template. To display the ISBN, create the column and enter `{identifiers:select(isbn)}` into the template box. To display a column containing the values of two series custom columns separated by a comma, use `{#series1:||,}{#series2}`.

Composite columns can use any template option, including formatting.

You cannot change the data contained in a composite column. If you edit a composite column by double-clicking on any item, you will open the template for editing, not the underlying data. Editing the template on the GUI is a quick way of testing and changing composite columns.

Using functions in templates - single-function mode

Suppose you want to display the value of a field in upper case, when that field is normally in title case. You can do this (and many more things) using the functions available for templates. For example, to display the title in upper case, use

[74] https://docs.python.org/2/library/string.html#format-string-syntax

`{title:uppercase()}`. To display it in title case, use `{title:titlecase()}`.

Function references appear in the format part, going after the `:` and before the first `|` or the closing `}`. If you have both a format and a function reference, the function comes after another `:`. Functions must always end with `()`. Some functions take extra values (arguments), and these go inside the `()`.

Functions are always applied before format specifications. See further down for an example of using both a format and a function, where this order is demonstrated.

The syntax for using functions is `{field:function(arguments)}`, or `{field:function(arguments)|prefix|suffix}`. Arguments are separated by commas. Commas inside arguments must be preceeded by a backslash ('\'). The last (or only) argument cannot contain a closing parenthesis (')'). Functions return the value of the field used in the template, suitably modified.

Important: If you have programming experience, please note that the syntax in this mode (single function) is not what you might expect. Strings are not quoted. Spaces are significant. All arguments must be constants; there is no sub-evaluation. **Do not use subtemplates ('{ ... }') as function arguments.** Instead, use *template program mode* (stránka 138) and *general program mode* (stránka 159).

Many functions use regular expressions. In all cases, regular expression matching is case-insensitive.

The functions available are listed below. Note that the definitive documentation for functions is available in the section *Function classification* (stránka 143):

- `lowercase()` – return value of the field in lower case.
- `uppercase()` – return the value of the field in upper case.
- `titlecase()` – return the value of the field in title case.
- `capitalize()` – return the value with the first letter upper case and the rest lower case.
- `contains(pattern, text if match, text if not match)` – checks if field contains matches for the regular expression *pattern*. Returns *text if match* if matches are found, otherwise it returns *text if no match*.
- `count(separator)` – interprets the value as a list of items separated by *separator*, returning the number of items in the list. Most lists use a comma as the separator, but authors uses an ampersand. Examples: *{tags:count(,)}*, *{authors:count(&)}*
- `format_number(template)` – interprets the value as a number and format that number using a python formatting template such as "{0:5.2f}" or "{0:,d}" or "${0:5,.2f}". The field_name part of the template must be a 0 (zero) (the "{0:" in the above examples). See the template language and python documentation for more examples. Returns the empty string if formatting fails.
- `human_readable()` – expects the value to be a number and returns a string representing that number in KB, MB, GB, etc.
- `ifempty(text)` – if the field is not empty, return the value of the field. Otherwise return *text*.
- `in_list(separator, pattern, found_val, not_found_val)` – interpret the field as a list of items separated by *separator*, comparing the *pattern* against each value in the list. If the pattern matches a value, return *found_val*, otherwise return *not_found_val*.
- `language_codes(lang_strings)` – return the language codes for the strings passed in *lang_strings*. The strings must be in the language of the current locale. *Lang_strings* is a comma-separated list.
- `language_strings(lang_codes, localize)` – return the strings for the language codes passed in *lang_codes*. If *localize* is zero, return the strings in English. If localize is not zero, return the strings in the language of the current locale. *Lang_codes* is a comma-separated list.
- `list_item(index, separator)` – interpret the field as a list of items separated by *separator*, returning the *index'th item. The first item is number zero. The last item can be returned using 'list_item(-1,separator)*. If

the item is not in the list, then the empty value is returned. The separator has the same meaning as in the *count* function.

- `lookup(pattern, field, pattern, field, ..., else_field)` – like switch, except the arguments are field (metadata) names, not text. The value of the appropriate field will be fetched and used. Note that because composite columns are fields, you can use this function in one composite field to use the value of some other composite field. This is extremely useful when constructing variable save paths (more later).
- `re(pattern, replacement)` – return the field after applying the regular expression. All instances of *pattern* are replaced with *replacement*. As in all of calibre, these are python-compatible regular expressions.
- `select(key)` – interpret the field as a comma-separated list of items, with the items being of the form "id:value". Find the pair with the id equal to key, and return the corresponding value. This function is particularly useful for extracting a value such as an isbn from the set of identifiers for a book.
- `shorten(left chars, middle text, right chars)` – Return a shortened version of the field, consisting of *left chars* characters from the beginning of the field, followed by *middle text*, followed by *right chars* characters from the end of the string. *Left chars* and *right chars* must be integers. For example, assume the title of the book is *Ancient English Laws in the Times of Ivanhoe*, and you want it to fit in a space of at most 15 characters. If you use `{title:shorten(9,-,5)}`, the result will be *Ancient E-nhoe*. If the field's length is less than `left chars` + `right chars` + the length of `middle text`, then the field will be used intact. For example, the title *The Dome* would not be changed.
- `str_in_list(val, separator, string, found_val, not_found_val)` – treat val as a list of items separated by separator, comparing the string against each value in the list. If the string matches a value, return found_val, otherwise return not_found_val. If the string contains separators, then it is also treated as a list and each value is checked.
- `subitems(val, start_index, end_index)` – This function is used to break apart lists of tag-like hierarchical items such as genres. It interprets the value as a comma-separated list of tag-like items, where each item is a period-separated list. Returns a new list made by first finding all the period-separated tag-like items, then for each such item extracting the components from *start_index* to *end_index*, then combining the results back together. The first component in a period-separated list has an index of zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list. Examples:

```
Assuming a #genre column containing "A.B.C":
    {#genre:subitems(0,1)} returns "A"
    {#genre:subitems(0,2)} returns "A.B"
    {#genre:subitems(1,0)} returns "B.C"
Assuming a #genre column containing "A.B.C, D.E":
    {#genre:subitems(0,1)} returns "A, D"
    {#genre:subitems(0,2)} returns "A.B, D.E"
```

- `sublist(val, start_index, end_index, separator)` – interpret the value as a list of items separated by *separator*, returning a new list made from the items from *start_index'to 'end_index*. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list. Examples assuming that the tags column (which is comma-separated) contains "A, B ,C":

```
{tags:sublist(0,1,\,)} returns "A"
{tags:sublist(-1,0,\,)} returns "C"
{tags:sublist(0,-1,\,)} returns "A, B"
```

- `swap_around_comma(val)` `` -- given a value of the form ``B, A`, return `A B`. This is most useful for converting names in LN, FN format to FN LN. If there is no comma, the function returns val unchanged.
- `switch(pattern, value, pattern, value, ..., else_value)` – for each `pattern`,

value pair, checks if the field matches the regular expression pattern and if so, returns that value. If no pattern matches, then else_value is returned. You can have as many pattern, value pairs as you want.

- test(text if not empty, text if empty) – return *text if not empty* if the field is not empty, otherwise return *text if empty*.
- transliterate() – Returns a string in a latin alphabet formed by approximating the sound of the words in the source field. For example, if the source field is Фёдор Михайлович Достоевский the function returns Fiodor Mikhailovich Dostoievskii.'

Now, what about using functions and formatting in the same field. Suppose you have an integer custom column called #myint that you want to see with leading zeros, as in 003. To do this, you would use a format of 0>3s. However, by default, if a number (integer or float) equals zero then the field produces the empty value, so zero values will produce nothing, not 000. If you really want to see 000 values, then you use both the format string and the ifempty function to change the empty value back to a zero. The field reference would be:

```
{#myint:0>3s:ifempty(0)}
```

Note that you can use the prefix and suffix as well. If you want the number to appear as [003] or [000], then use the field:

```
{#myint:0>3s:ifempty(0)|[|]}
```

Using functions in templates - template program mode

The template language program mode differs from single-function mode in that it permits you to write template expressions that refer to other metadata fields, modify values, and do arithmetic. It is a reasonably complete programming language.

You can use the functions documented above in template program mode. See below for details.

Beginning with an example, assume that you want your template to show the series for a book if it has one, otherwise show the value of a custom field #genre. You cannot do this in the basic language because you cannot make reference to another metadata field within a template expression. In program mode, you can. The following expression works:

```
{#series:'ifempty($, field('#genre'))'}
```

The example shows several things:

- program mode is used if the expression begins with :' and ends with '. Anything else is assumed to be single-function.
- the variable $ stands for the field the expression is operating upon, #series in this case.
- functions must be given all their arguments. There is no default value. For example, the standard built-in functions must be given an additional initial parameter indicating the source field, which is a significant difference from single-function mode.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either ' or ".

The language is similar to functional languages in that it is built almost entirely from functions. A statement is a function. An expression is a function. Constants and identifiers can be thought of as functions returning the value indicated by the constant or stored in the identifier.

The syntax of the language is shown by the following grammar:

```
constant   ::= " string " | ' string ' | number
identifier ::= sequence of letters or ``_`` characters
function   ::= identifier ( statement [ , statement ]* )
expression ::= identifier | constant | function | assignment
assignment ::= identifier '=' expression
statement  ::= expression [ ; expression ]*
program    ::= statement
```

Comments are lines with a '#' character at the beginning of the line.

An `expression` always has a value, either the value of the constant, the value contained in the identifier, or the value returned by a function. The value of a `statement` is the value of the last expression in the sequence of statements. As such, the value of the program (statement):

```
1; 2; 'foobar'; 3
```

is 3.

Another example of a complex but rather silly program might help make things clearer:

```
{series_index:'
    substr(
        strcat($, '->',
            cmp(divide($, 2), 1,
                assign(c, 1); substr('lt123', c, 0),
                'eq', 'gt')),
        0, 6)
   '| prefix | suffix}
```

This program does the following:

- specify that the field being looked at is series_index. This sets the value of the variable `$`.
- calls the `substr` function, which takes 3 parameters `(str, start, end)`. It returns a string formed by extracting the start through end characters from string, zero-based (the first character is character zero). In this case the string will be computed by the `strcat` function, the start is 0, and the end is 6. In this case it will return the first 6 characters of the string returned by `strcat`, which must be evaluated before substr can return.
- calls the `strcat` (string concatenation) function. Strcat accepts 1 or more arguments, and returns a string formed by concatenating all the values. In this case there are three arguments. The first parameter is the value in `$`, which here is the value of `series_index`. The second paremeter is the constant string `'->'`. The third parameter is the value returned by the `cmp` function, which must be fully evaluated before `strcat` can return.
- The `cmp` function takes 5 arguments `(x, y, lt, eq, gt)`. It compares x and y and returns the third argument `lt` if x < y, the fourth argument `eq` if x == y, and the fifth argument `gt` if x > y. As with all functions, all of the parameters can be statements. In this case the first parameter (the value for `x`) is the result of dividing the series_index by 2. The second parameter `y` is the constant `1`. The third parameter `lt` is a statement (more later). The fourth parameter `eq` is the constant string `'eq'`. The fifth parameter is the constant string `'gt'`.
- The third parameter (the one for `lt`) is a statement, or a sequence of expressions. Remember that a statement (a sequence of semicolon-separated expressions) is also an expression, returning the value of the last expression in the list. In this case, the program first assigns the value `1` to a local variable `c`, then returns a substring made by extracting the c'th character to the end. Since c always contains the constant `1`, the substring will return the second through end'th characters, or `'t123'`.
- Once the statement providing the value to the third parameter is executed, `cmp` can return a value. At that point, `strcat` can return a value, then ``substr`` can return a value. The program then terminates.

For various values of series_index, the program returns:

- series_index == undefined, result = `prefix ->t123 suffix`

- series_index == 0.5, result = `prefix 0.50-> suffix`
- series_index == 1, result = `prefix 1->t12 suffix`
- series_index == 2, result = `prefix 2->eq suffix`
- series_index == 3, result = `prefix 3->gt suffix`

All the functions listed under single-function mode can be used in program mode. To do so, you must supply the value that the function is to act upon as the first parameter, in addition to the parameters documented above. For example, in program mode the parameters of the *test* function are `test(x, text_if_not_empty, text_if_empty)`. The *x* parameter, which is the value to be tested, will almost always be a variable or a function call, often *field()*.

The following functions are available in addition to those described in single-function mode. Remember from the example above that the single-function mode functions require an additional first parameter specifying the field to operate on. With the exception of the `id` parameter of assign, all parameters can be statements (sequences of expressions). Note that the definitive documentation for functions is available in the section *Function classification* (stránka 143):

- `and(value, value, ...)` – returns the string "1" if all values are not empty, otherwise returns the empty string. This function works well with test or first_non_empty. You can have as many values as you want.
- `add(x, y)` – returns x + y. Throws an exception if either x or y are not numbers.
- `assign(id, val)` – assigns val to id, then returns val. id must be an identifier, not an expression
- `approximate_formats()` – return a comma-separated list of formats that at one point were associated with the book. There is no guarantee that the list is correct, although it probably is. This function can be called in template program mode using the template `{:'approximate_formats()'}`. Note that format names are always uppercase, as in EPUB.
- `author_links(val_separator, pair_separator)` – returns a string containing a list of authors and that author's link values in the form `author1 val_separator author1link pair_separator author2 val_separator author2link` etc. An author is separated from its link value by the `val_separator` string with no added spaces. `author:linkvalue` pairs are separated by the `pair_separator` string argument with no added spaces. It is up to you to choose separator strings that do not occur in author names or links. An author is included even if the author link is empty.
- `author_sorts(val_separator)` – returns a string containing a list of author's sort values for the authors of the book. The sort is the one in the author metadata (different from the author_sort in books). The returned list has the form author sort 1 `val_separator` author sort 2 etc. The author sort values in this list are in the same order as the authors of the book. If you want spaces around `val_separator` then include them in the separator string
- `booksize()` – returns the value of the calibre 'size' field. Returns '' if there are no formats.
- `cmp(x, y, lt, eq, gt)` – compares x and y after converting both to numbers. Returns `lt` if x < y. Returns `eq` if x == y. Otherwise returns `gt`.
- `current_library_name() -- `` return the last name on the path to the current calibre library. This function can be called in template program mode using the template ``{:'current_library_name()'}.`
- `current_library_path() -- `` return the path to the current calibre library. This function can be called in template program mode using the template ``{:'current_library_path()'}..`
- `days_between(date1, date2)` – return the number of days between `date1` and `date2`. The number is positive if `date1` is greater than `date2`, otherwise negative. If either `date1` or `date2` are not dates, the function returns the empty string.
- `divide(x, y)` – returns x / y. Throws an exception if either x or y are not numbers.

- `eval(string)` – evaluates the string as a program, passing the local variables (those `assign` ed to). This permits using the template processor to construct complex results from local variables. Because the *{* and *}* characters are special, you must use *[[* for the *{* character and *]]* for the '}' character; they are converted automatically. Note also that prefixes and suffixes (the *|prefix|suffix* syntax) cannot be used in the argument to this function when using template program mode.
- `field(name)` – returns the metadata field named by `name`.
- `first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result)` – compares `val < cmpN` in sequence, returning resultN for the first comparison that succeeds. Returns else_result if no comparison succeeds. Example:

```
first_matching_cmp(10,5,"small",10,"middle",15,"large","giant")
```

returns "large". The same example with a first value of 16 returns "giant".

- `first_non_empty(value, value, ...)` – returns the first value that is not empty. If all values are empty, then the empty value is returned. You can have as many values as you want.
- `format_date(x, date_format)` – format the value, which must be a date field, using the format_string, returning a string. The formatting codes are:

```
d    : the day as number without a leading zero (1 to 31)
dd   : the day as number with a leading zero (01 to 31)
ddd  : the abbreviated localized day name (e.g. "Mon" to "Sun").
dddd : the long localized day name (e.g. "Monday" to "Sunday").
M    : the month as number without a leading zero (1 to 12).
MM   : the month as number with a leading zero (01 to 12)
MMM  : the abbreviated localized month name (e.g. "Jan" to "Dec").
MMMM : the long localized month name (e.g. "January" to "December").
yy   : the year as two digit number (00 to 99).
yyyy : the year as four digit number.
h    : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
hh   : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
m    : the minutes without a leading 0 (0 to 59)
mm   : the minutes with a leading 0 (00 to 59)
s    : the seconds without a leading 0 (0 to 59)
ss   : the seconds with a leading 0 (00 to 59)
ap   : use a 12-hour clock instead of a 24-hour clock, with 'ap' replaced by the localized strin
AP   : use a 12-hour clock instead of a 24-hour clock, with 'AP' replaced by the localized strin
iso  : the date with time and timezone. Must be the only format present.
```

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the format tweaks to contain `MMMM`. In this case, instead of using something like `{pubdate:format_date(yyyy)}`, write the template using template program mode as in `{:'format_date(raw_field('pubdate'),'yyyy')'}`.

- `finish_formatting(val, fmt, prefix, suffix)` – apply the format, prefix, and suffix to a value in the same way as done in a template like `{series_index:05.2f|- |- }`. This function is provided to ease conversion of complex single-function- or template-program-mode templates to *general program mode* (stránka 159) (see below) to take advantage of GPM template compilation. For example, the following program produces the same output as the above template:

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

Another example: for the template `{series:re(([^\s])[^\s]+(\s|$),\1)}{series_index:0>2s|- | - }{title}` use:

```
program:
    strcat(
```

```
            re(field('series'), '([^\s])[^\s]+(\s|$)', '\1'),
            finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
            field('title')
        )
```

- `formats_modtimes(date_format)` – return a comma-separated list of colon-separated items representing modification times for the formats of a book. The date_format parameter specifies how the date is to be formatted. See the date_format function for details. You can use the select function to get the mod time for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_paths()` – return a comma-separated list of colon-separated items representing full path to the formats of a book. You can use the select function to get the path for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_sizes()` – return a comma-separated list of colon-separated items representing sizes in bytes of the formats of a book. You can use the select function to get the size for a specific format. Note that format names are always uppercase, as in EPUB.
- `has_cover()` – return `Yes` if the book has a cover, otherwise return the empty string
- `not(value)` – returns the string "1" if the value is empty, otherwise returns the empty string. This function works well with test or first_non_empty. You can have as many values as you want.
- `list_difference(list1, list2, separator)` – return a list made by removing from *list1* any item found in *list2*, using a case-insensitive compare. The items in *list1* and *list2* are separated by separator, as are the items in the returned list.
- `list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – return *yes_val* if *list1* and *list2* contain the same items, otherwise return *no_val*. The items are determined by splitting each list using the appropriate separator character (*sep1* or *sep2*). The order of items in the lists is not relevant. The compare is case insensitive.
- `list_intersection(list1, list2, separator)` – return a list made by removing from *list1* any item not found in *list2*, using a case-insensitive compare. The items in *list1* and *list2* are separated by separator, as are the items in the returned list.
- `list_re(src_list, separator, include_re, opt_replace)` – Construct a list by first separating *src_list* into items using the *separator* character. For each item in the list, check if it matches *include_re*. If it does, then add it to the list to be returned. If *opt_replace* is not the empty string, then apply the replacement before adding the item to the returned list.
- `list_re_group(src_list, separator, include_re, search_re, template_for_group_1, for_group_2, ...)` – Like list_re except replacements are not optional. It uses re_group(item, search_re, template ...) when doing the replacements.
- `list_sort(list, direction, separator)` – return list sorted using a case-insensitive sort. If *direction* is zero, the list is sorted ascending, otherwise descending. The list items are separated by separator, as are the items in the returned list.
- `list_union(list1, list2, separator)` – return a list made by merging the items in list1 and list2, removing duplicate items using a case-insensitive compare. If items differ in case, the one in list1 is used. The items in list1 and list2 are separated by separator, as are the items in the returned list.
- `multiply(x, y)` – returns x * y. Throws an exception if either x or y are not numbers.
- `ondevice()` – return the string "Yes" if ondevice is set, otherwise return the empty string
- `or(value, value, ...)` – returns the string "1" if any value is not empty, otherwise returns the empty string. This function works well with test or first_non_empty. You can have as many values as you want.

- `print(a, b, ...)` – prints the arguments to standard output. Unless you start calibre from the command line (`calibre-debug -g`), the output will go to a black hole.
- `raw_field(name)` – returns the metadata field named by name without applying any formatting.
- `raw_list(name, separator)` – returns the metadata list named by name without applying any formatting or sorting and with items separated by separator.
- `re_group(val, pattern, template_for_group_1, for_group_2, ...)` – return a string made by applying the regular expression pattern to the val and replacing each matched instance with the string computed by replacing each matched group by the value returned by the corresponding template. The original matched value for the group is available as $. In template program mode, like for the template and the eval functions, you use [[for { and]] for }. The following example in template program mode looks for series with more than one word and uppercases the first word:

```
{series:'re_group($, "(\S* )(.*)", "[[$:uppercase()]]", "[[$]]")'}
```

- `series_sort()` – returns the series sort value.
- `strcat(a, b, ...)` – can take any number of arguments. Returns a string formed by concatenating all the arguments.
- `strcat_max(max, string1, prefix2, string2, ...)` – Returns a string formed by concatenating the arguments. The returned value is initialized to string1. *Prefix, string* pairs are added to the end of the value as long as the resulting string length is less than *max*. String1 is returned even if string1 is longer than max. You can pass as many *prefix, string* pairs as you wish.
- `strcmp(x, y, lt, eq, gt)` – does a case-insensitive comparison x and y as strings. Returns `lt` if x < y. Returns `eq` if x == y. Otherwise returns `gt`.
- `strlen(a)` – Returns the length of the string passed as the argument.
- `substr(str, start, end)` – returns the `start`'th through the `end`'th characters of `str`. The first character in `str` is the zero'th character. If end is negative, then it indicates that many characters counting from the right. If end is zero, then it indicates the last character. For example, `substr('12345', 1, 0)` returns `'2345'`, and `substr('12345', 1, -1)` returns `'234'`.
- `subtract(x, y)` – returns x - y. Throws an exception if either x or y are not numbers.
- `today()` – return a date string for today. This value is designed for use in format_date or days_between, but can be manipulated like any other string. The date is in ISO format.
- `template(x)` – evaluates x as a template. The evaluation is done in its own context, meaning that variables are not shared between the caller and the template evaluation. Because the *{* and *}* characters are special, you must use *[[* for the *{* character and *]]* for the '}' character; they are converted automatically. For example, `template('[[title_sort]]') will evaluate the template ''{title_sort}` and return its value. Note also that prefixes and suffixes (the *|prefix|suffix* syntax) cannot be used in the argument to this function when using template program mode.

Function classification

Reference for all built-in template language functions

Here, we document all the built-in functions available in the calibre template language. Every function is implemented as a class in python and you can click the source links to see the source code, in case the documentation is insufficient. The functions are arranged in logical groups by type.

- *Arithmetic* (stránka 146)
  - *add(x, y)* (stránka 146)
  - *divide(x, y)* (stránka 146)
  - *multiply(x, y)* (stránka 146)
  - *subtract(x, y)* (stránka 146)
- *Boolean* (stránka 146)
  - *and(hodnota, hodnota, ...)* (stránka 146)
  - *nebo(hodnota, hodnota, ...)* (stránka 146)
  - *not(hodnota)* (stránka 146)
- *Date functions* (stránka 146)
  - *days_between(datum1, datum2)* (stránka 146)
  - *today()* (stránka 146)
- *Formatting values* (stránka 147)
  - *finish_formatting(hodnota, formát, předpona, přípona)* (stránka 147)
  - *format_date(hodnota, formát_řetězce)* (stránka 147)
  - *format_number(hodnota, šablona)* (stránka 147)
  - *human_readable(hodnota)* (stránka 147)
- *Get values from metadata* (stránka 147)
  - *approximate_formats()* (stránka 147)
  - *author_links(val_separator, pair_separator)* (stránka 147)
  - *author_sorts(val_separator)* (stránka 148)
  - *booksize()* (stránka 148)
  - *current_library_name()* (stránka 148)
  - *current_library_path()* (stránka 148)
  - *field(name)* (stránka 148)
  - *formats_modtimes(formát_data)* (stránka 148)
  - *formats_paths()* (stránka 148)
  - *formats_sizes()* (stránka 149)
  - *has_cover()* (stránka 149)
  - *language_codes(řetězce_jazyka)* (stránka 149)
  - *language_strings(kódy_jazyka, lokalizovat)* (stránka 149)
  - *ondevice()* (stránka 149)
  - *raw_field(name)* (stránka 149)
  - *raw_list(název, oddělovač)* (stránka 149)
  - *series_sort()* (stránka 149)
  - *user_categories()* (stránka 149)
  - *virtual_libraries()* (stránka 150)
- *If-then-else* (stránka 150)
  - *contains(hodnota, vzor, text při shodě, text při neshodě)* (stránka 150)
  - *ifempty(hodnota, text při prázdné)* (stránka 150)
  - *test(hodnota, text při neprázdném, text při prázdném)* (stránka 150)
- *Iterating over values* (stránka 150)
  - *first_non_empty(hodnota, hodnota, ...)* (stránka 150)
  - *lookup(hodnota, vzor, pole, vzor, pole, ..., jinak_pole)* (stránka 150)
  - *switch(hodnota, vzor, hodnota, vzor, hodnota, ..., jinak_hodnota)* (stránka 150)
- *List lookup* (stránka 150)
  - *identifier_in_list(hodnota, identifikátor, nalezená_hodnota, nenalezená_hodnota)* (stránka 151)
  - *in_list(hodnota, oddělovač, vzor, nalezená_hodnota, nenalezená_hodnota)* (stránka 151)
  - *list_item(hodnota, pořadí, oddělovač)* (stránka 151)
  - *select(hodnota, klíč)* (stránka 151)
  - *str_in_list(hodnota, oddělovač, řetězec, nalezená_hodnota, nenalezená_hodnota)* (stránka 151)
- *List manipulation* (stránka 151)
  - *count(hodnota, oddělovač)* (stránka 151)
  - *list_difference(seznam1, seznam2, oddělovač)* (stránka 151)
  - *list_equals(seznam1, oddělovač1, seznam2, oddělovač2, hodnota_ano, hodnota_ne)* (stránka 152)
  - *list_intersection(seznam1, seznam2, oddělovač)* (stránka 152)
  - *list_re(seznam_zdrojů, oddělovač, zahrnout_vrácené, volitelně_nahradit)* (stránka 152)
  - *list_re_group(src_list, separator, include_re, search_re, group_1_template, ...)* (stránka 152)
  - *list_sort(seznam, směr, oddělovač)* (stránka 152)
  - *list_union(seznam1, seznam2, oddělovač)* (stránka 152)

Arithmetic

add(x, y)
class calibre.utils.formatter_functions.**BuiltinAdd**
add(x, y) – vrací x + y. Vyvolá výjimku, když buď x nebo y nejsou čísla.

divide(x, y)
class calibre.utils.formatter_functions.**BuiltinDivide**
divide(x, y) – vrací x / y. Vyvolá výjimku, když buď x nebo y nejsou čísla.

multiply(x, y)
class calibre.utils.formatter_functions.**BuiltinMultiply**
multiply(x, y) – vrací x * y. Vyvolá výjimku, když buď x nebo y nejsou čísla.

subtract(x, y)
class calibre.utils.formatter_functions.**BuiltinSubtract**
subtract(x, y) – vrací x - y. Vyvolá výjimku, když buď x nebo y nejsou čísla.

Boolean

and(hodnota, hodnota, ...)
class calibre.utils.formatter_functions.**BuiltinAnd**
and(hodnota, hodnota, ...) – vrací řetězec "1", pokud všechny hodnoty nejsou prázdné, jinak vrací prázdný řetězec. Tato funkce funguje dobře s test nebo first_non_empty. Můžete mít tolik hodnot, kolik chcete.

nebo(hodnota, hodnota, ...)
class calibre.utils.formatter_functions.**BuiltinOr**
nebo(hodnota, hodnota, ...) – vrací řetězec "1", pokud jakákoliv hodnota není prázdná, jinak vrací prázdný řetězec. Tato funkce funguje dobře s test nebo first_non_empty. Můžete mít tolik hodnot, kolik chcete.

not(hodnota)
class calibre.utils.formatter_functions.**BuiltinNot**
not(hodnota) – vrací řetězec "1", pokud je hodnota prázdná, jinak vrací prázdný řetězec. Tato funkce funguje dobře s test nebo first_non_empty. Můžete mít tolik hodnot, kolik chcete.

Date functions

days_between(datum1, datum2)
class calibre.utils.formatter_functions.**BuiltinDaysBetween**
days_between(datum1, datum2) – vrací počet dní mezi datum1 a datum2. Počet je kladný, pokud je datum1 větší než datum2, jinak je záporný. Pokud datum1 nebo datum2 nejsou data, funkce vrací prázdný řetězec.

today()
class calibre.utils.formatter_functions.**BuiltinToday**
today() – vrací řetězec data pro dnešek. Tato hodnota je navržena pro použití ve format_date nebo days_between, ale může s ní být zacházeno jako s jakýmkoliv jiným řetězcem. Datum je ve formátu ISO.

Formatting values

finish_formatting(hodnota, formát, předpona, přípona)

class calibre.utils.formatter_functions.**BuiltinFinishFormatting**

finish_formatting(hodnota, formát, předpona, přípona) – použije formát, předponu a příponu na hodnotu stejným způsobem, jak bylo provedeno v šabloně jako *{series_index:05.2f| - |- }*. Například následující vytváří stejný výstup jako výše uvedená šablona: program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")

format_date(hodnota, formát_řetězce)

class calibre.utils.formatter_functions.**BuiltinFormatDate**

format_date(hodnota, formát_řetězce) – formátuje hodnotu, což musí být datum, pomocí formát_řetězce, vrací řetězec. Kódy formátování jsou: d: den jako číslo bez úvodní nuly (1 až 31), dd: den jako číslo s úvodní nulou (01 až 31), ddd: zkrácený lokalizovaný název dne (např. "Pon" až "Ned"), dddd: dlouhý lokalizovaný název dne (např. "Pondělí" až "Neděle"). M: měsíc jako číslo bez úvodní nuly (1 až 12), MM: měsíc jako číslo s úvodní nulou (01 až 12), MMM: zkrácený lokalizovaný název měsíce (např. "Led" až "Pro"), MMMM: dlouhý lokalizovaný název měsíce (např. "Leden" až "Prosinec"). yy: rok jako dvoumístné číslo (00 až 99), yyyy: rok jako čtyřmístné číslo. h: hodiny bez úvodní nuly (0 až 11 nebo 0 až 23, v závislosti na dop./odp.), hh: hodiny s úvodní nulou (00 až 11 nebo 00 až 23, v závislosti na dop./odp.), m: minuty bez úvodní nuly (0 až 59), mm: minuty s úvodní nulou (00 až 59), s: sekundy bez úvodní nuly (0 až 59), ss: sekundy s úvodní nulou (00 až 59). ap: použít 12hodinový formát namísto 24hodinového formátu, kde je "ap" nahrazeno lokalizovaným řetězcem pro am nebo pm, AP: použít 12hodinový formát namísto 24hodinového formátu, kde je "AP" nahrazeno lokalizovaným řetězcem pro AM nebo PM, iso: datum s časem a časovou zónou. Musí být přítomen jediný formát

format_number(hodnota, šablona)

class calibre.utils.formatter_functions.**BuiltinFormatNumber**

format_number(hodnota, šablona) – formátuje číslo hodnota pomocí šablony formátování Pythonu, jako je "{0:5.2f}" nebo "{0:,d}" nebo "${0:5,.2f}". Součást šablony název_pole musí být 0 (nula) ("{0:" ve výše uvedených příkladech). Pro více příkladů se podívejte na jazyk šablony a do dokumentace Pythonu. Vrací prázdný řetězec, pokud formátování selže.

human_readable(hodnota)

class calibre.utils.formatter_functions.**BuiltinHumanReadable**

human_readable(hodnota) – vrací řetězec představující číslo v KB, MB, GB atd.

Get values from metadata

approximate_formats()

class calibre.utils.formatter_functions.**BuiltinApproximateFormats**

approximate_formats() – vrací čárkami oddělený seznam formátů, které svého času byly přidruženy ke knize. Není žádná záruka, že je tento seznam správný, ačkoliv nejspíš je. Tato funkce může být volána v režimu šablony programu pomocí šablony "{:'approximate_formats()'}". Pamatujte, že názvy formátů jsou vždy velkými písmeny, jako v EPUB. Tato funkce funguje pouze v grafickém rozhraní. Pokud chcete použít tyto hodnoty v šablonách uložení na disk nebo odeslání do zařízení, pak musíte vytvořit vlastní "Sloupec sestavený z jiných sloupců", použít funkci v šabloně toho sloupce a použít hodnotu toho sloupce ve své šabloně uložení nebo odeslání

author_links(val_separator, pair_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**
author_links(val_separator, pair_separator) – vrací řetězec obsahující seznam autorů a hodnoty odkazů na tyto autory ve formě author1 val_separator author1link pair_separator author2 val_separator author2link atd. Autor je oddělen od své hodnoty odkazu řetězcem val_separator bez přidaných mezer. Páry author:linkvalue jsou odděleny parametrem řetězce pair_separator bez přidaných mezer. Je na vás zvolit si řetězce oddělovače, které se nevyskytují ve jménech autorů nebo odkazech. Autor je zahrnut i v případě, že odkaz na autora je prázdný.

author_sorts(val_separator)
class calibre.utils.formatter_functions.**BuiltinAuthorSorts**
author_sorts(val_separator) – vrací řetězec obsahující seznam hodnot řazení autorů pro autory knihy. Řazení je takové jako v metadatech autora (odlišné od author_sort v knihách). Vrácený seznam má podobu řazení autora 1 val_separator řazení autora 2 atd. Hodnoty řazení autora v tomto seznamu jsou ve stejném pořadí jako autoři knihy. Pokud chcete kolem val_separator mezery, tak je zahrňte do řetězce oddělovače

booksize()
class calibre.utils.formatter_functions.**BuiltinBooksize**
booksize() – vrací hodnotu pole velikost. Tato funkce funguje pouze v grafickém rozhraní. Pokud chcete použít tuto hodnotu v šablonách uložení na disk nebo odeslání do zařízení, pak musíte vytvořit vlastní "Sloupec sestavený z jiných sloupců", použít funkci v šabloně toho sloupce a použít hodnotu toho sloupce ve své šabloně uložení nebo odeslání

current_library_name()
class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**
current_library_name() – vrací poslední název v cestě k aktuální knihovně Calibre. Tato funkce může být volána v režimu šablony programu pomocí šablony "{:'current_library_name()'}".

current_library_path()
class calibre.utils.formatter_functions.**BuiltinCurrentLibraryPath**
current_library_path() – vrací cestu k aktuální knihovně Calibre. Tato funkce může být volána v režimu šablony programu pomocí šablony "{:'current_library_path()'}".

field(name)
class calibre.utils.formatter_functions.**BuiltinField**
field(name) – vrací pole metadata pojmenované podle name

formats_modtimes(formát_data)
class calibre.utils.formatter_functions.**BuiltinFormatsModtimes**
formats_modtimes(formát_data) – vrací čárkami oddělený seznam dvojtečkami oddělených položek představujících časy úpravy pro formáty knihy. Parametr formát_data určuje, jak má být datum formátováno. Pro podrobnosti se podívejte na funkci date_format. Můžete použít funkci select pro získání času úpravy konkrétního formátu. Pamatujte, že názvy formátu jsou vždy velkými písmeny, jako v EPUB.

formats_paths()
class calibre.utils.formatter_functions.**BuiltinFormatsPaths**
formats_paths() – vrací čárkami oddělený seznam dvojtečkami oddělených položek představujících celé cesty k formátům knihy. Můžete použít funkci select pro získání cesty ke konkrétnímu formátu. Pamatujte, že názvy formátu jsou vždy velkými písmeny, jako v EPUB.

formats_sizes()
class calibre.utils.formatter_functions.**BuiltinFormatsSizes**
formats_sizes() – vrací čárkami oddělený seznam dvojtečkami oddělených položek představujících velikosti formátů knihy v bajtech. Můžete použít funkci select pro získání velikosti pro konkrétní formát. Pamatujte, že názvy formátu jsou vždy velkými písmeny, jako v EPUB.

has_cover()
class calibre.utils.formatter_functions.**BuiltinHasCover**
has_cover() – vrací Ano, pokud má kniha obálku, jinak vrací prázdný řetězec

language_codes(řetězce_jazyka)
class calibre.utils.formatter_functions.**BuiltinLanguageCodes**
language_codes(řetězce_jazyka) – vrací kódy jazyka pro řetězce předané v řetězce_jazyka. Řetězce musí být v jazyce aktuálního národního prostředí. Řetězce_jazyka je čárkami oddělený seznam.

language_strings(kódy_jazyka, lokalizovat)
class calibre.utils.formatter_functions.**BuiltinLanguageStrings**
language_strings(kódy_jazyka, lokalizovat) – vrací řetězec pro kódy jazyka předané v kódy_jazyka. Pokud lokalizovat je nula, vrací řetězce v angličtině. Pokud lokalizovat není nula, vrací řetězce v jazyce aktuálního národního prostředí. Kódy_jazyka je čárkami oddělený seznam.

ondevice()
class calibre.utils.formatter_functions.**BuiltinOndevice**
ondevice() – vrací Ano, pokud je nastaveno ondevice, jinak vrací prázdný řetězec. Tato funkce funguje pouze v grafickém rozhraní. Pokud chcete použít tuto hodnotu v šablonách uložení na disk nebo odeslání do zařízení, pak musíte vytvořit vlastní "Sloupec sestavený z jiných sloupců", použít funkci v šabloně toho sloupce a použít hodnotu toho sloupce ve své šabloně uložení nebo odeslání

raw_field(name)
class calibre.utils.formatter_functions.**BuiltinRawField**
raw_field(name) – vrací pole metadata pojmenované podle name bez použití jakéhokoliv formátování.

raw_list(název, oddělovač)
class calibre.utils.formatter_functions.**BuiltinRawList**
raw_list(název, oddělovač) – vrací seznam metadat pojmenovaný názvem bez použití jakéhokoliv formátování nebo řazení a s položkami oddělenými oddělovačem.

series_sort()
class calibre.utils.formatter_functions.**BuiltinSeriesSort**
series_sort() – vrací hodnotu řazení série

user_categories()
class calibre.utils.formatter_functions.**BuiltinUserCategories**
user_categories() – vrátí čárkami oddělený seznam uživatelských kategorií, které obsahují tuto knihu. Tato funkce funguje pouze v grafickém rozhraníI. Pokud chcete použít tyto hodnoty v šablonách uložení na disk nebo odeslání do zařízení, pak musíte vytvořit vlastní "Sloupec sestavený z jiných sloupců", použít funkci v šabloně toho sloupce a použít hodnotu toho sloupce ve svých šablonách uložení nebo odeslání

virtual_libraries()

class calibre.utils.formatter_functions.**BuiltinVirtualLibraries**

virtual_libraries() – vrací čárkami oddělený seznam virtuálních knihoven, které obsahují tuto knihu. Tato funkce funguje pouze v grafickém rozhraní. Pokud chcete použít tyto hodnoty v šablonách uložení na disk nebo odeslání do zařízení, pak musíte vytvořit vlastní "Sloupec sestavený z jiných sloupců", použít funkci v šabloně toho sloupce a použít hodnotu toho sloupce ve svých šablonách uložení nebo odeslání

If-then-else

contains(hodnota, vzor, text při shodě, text při neshodě)

class calibre.utils.formatter_functions.**BuiltinContains**

contains(hodnota, vzor, text při shodě, text při neshodě) – kontroluje, zda pole obsahuje shody pro *vzor* regulárního výrazu. Vrací *text při shodě*, pokud jsou nalzeny shody, jinak vrací *text při neshodě*

ifempty(hodnota, text při prázdné)

class calibre.utils.formatter_functions.**BuiltinIfempty**

ifempty(hodnota, text při prázdné) – vrací hodnotu, pokud hodnota není prázdná, jinak vrací *text při prázdné*

test(hodnota, text při neprázdném, text při prázdném)

class calibre.utils.formatter_functions.**BuiltinTest**

test(hodnota, text při neprázdném, text při prázdném) – vrací *text při neprázdném*, pokud pole není prázdné, jinak vrací *text při prázdném*

Iterating over values

first_non_empty(hodnota, hodnota, ...)

class calibre.utils.formatter_functions.**BuiltinFirstNonEmpty**

first_non_empty(hodnota, hodnota, ...) – vrací první hodnotu, která není prázdná. Pokud jsou všechny hodnoty prázdné, pak je vrácena prázdná hodnota. Můžete mít tolik hodnot, kolik chcete.

lookup(hodnota, vzor, pole, vzor, pole, ..., jinak_pole)

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(hodnota, vzor, pole, vzor, pole, ..., jinak_pole) – jako switch kromě toho, že parametry jsou názvy polí (metadat), ne text. Hodnota vhodného pole bude načtena a použita. Pamatujte, že protože složené sloupce jsou pole, můžete použít tuto funkci v jednom složeném poli, aby použilo hodnotu některého jiného složeného pole. Toto je neobyčejně užitečné při sestavování proměnných cest uložení.

switch(hodnota, vzor, hodnota, vzor, hodnota, ..., jinak_hodnota)

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(hodnota, vzor, hodnota, vzor, hodnota, ..., jinak_hodnota) – pro každý pár *vzor, hodnota* kontroluje, zda pole odpovídá regulárnímu výrazu *vzor* a pokud ano, vrací tu *hodnotu*. Pokud neodpovídají žádné výrazy, pak je vrácena jinak_hodnota. Můžete mít tolik párů *vzor, hodnota*, kolik chcete.

List lookup

identifier_in_list(hodnota, identifikátor, nalezená_hodnota, nenalezená_hodnota)

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

identifier_in_list(hodnota, identifikátor, nalezená_hodnota, nenalezená_hodnota) – považuje hodnotu za seznam identifikátorů oddělených čárkami, porovnává řetězec proti každé hodnotě v seznamu. Identifikátor má formát "identifikátor:hodnota". Parametr identifikátor by měl být buď "id" nebo "id:regexp". První případ se shoduje, pokud je jakýkoliv identifikátor s tímto id. Druhý případ se shoduje, pokud regexp odpovídá hodnotě identifikátoru. Pokud je nějaká shoda, vrací nalezená_hodnota, jinak vrací nenalezená_hodnota.

in_list(hodnota, oddělovač, vzor, nalezená_hodnota, nenalezená_hodnota)

class calibre.utils.formatter_functions.**BuiltinInList**

in_list(hodnota, oddělovač, vzor, nalezená_hodnota, nenalezená_hodnota) – považuje hodnotu za seznam položek oddělených oddělovačem, porovnává vzor proti každé hodnotě v seznamu. Pokud vzor odpovídá hodnotě, vrací nalezená_hodnota, jinak vrací nenalezená_hodnota.

list_item(hodnota, pořadí, oddělovač)

class calibre.utils.formatter_functions.**BuiltinListitem**

list_item(hodnota, pořadí, oddělovač) – interpretuje hodnotu jako seznam položek oddělených *oddělovačem*, vrací položku na *pořadí*. První položka je číslo nula. Poslední položka může být vrácena pomocí *list_item(-1,oddělovač)*. Pokud položka naní v seznamu, pak je vrácena prázdné hodnota. Oddělovač má stejný význam jako ve funkci count.

select(hodnota, klíč)

class calibre.utils.formatter_functions.**BuiltinSelect**

select(hodnota, klíč) – interpretuje hodnotu jako čárkami oddělený seznam položek, s položkami "identifikátor:hodnota". Nejde pár s identifikátorem rovnajícím se klíči a vrátí odpovídající hodnotu.

str_in_list(hodnota, oddělovač, řetězec, nalezená_hodnota, nenalezená_hodnota)

class calibre.utils.formatter_functions.**BuiltinStrInList**

str_in_list(hodnota, oddělovač, řetězec, nalezená_hodnota, nenalezená_hodnota) – považuje hodnotu za seznam položek oddělených oddělovačem, porovnává řetězec proti každé hodnotě v seznamu. Pokud řetězec odpovídá hodnotě, vrací nalezená_hodnota, jinak vrací nenalezená_hodnota. Pokud řetězec obsahuje oddělovač, je také považován za seznam a je zkontrolována každá hodnota.

List manipulation

count(hodnota, oddělovač)

class calibre.utils.formatter_functions.**BuiltinCount**

count(hodnota, oddělovač) – interpretuje hodnotu jako seznam položek oddělených *oddělovačem*, vrací počet položek v seznamu. Většina seznamů používá jako oddělovač čárku, ale autoři používají znak &. Příklady: {tags:count(,)}, {authors:count(&)}

list_difference(seznam1, seznam2, oddělovač)

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(seznam1, seznam2, oddělovač) – vrací seznam vytvořený odebrání ze seznam1 všech položek nalezených v seznam2 pomocí porovnání bez rozlišení malých a velkých písmen. Položky v seznam1 a seznam2 jsou odděleny oddělovačem, stejně jako položky ve vráceném seznamu.

list_equals(seznam1, oddělovač1, seznam2, oddělovač2, hodnota_ano, hodnota_ne)

class calibre.utils.formatter_functions.**BuiltinListEquals**

list_equals(seznam1, oddělovač1, seznam2, oddělovač2, hodnota_ano, hodnota_ne) – vrací hodnota_ano, pokud seznam1 a seznam2 obsahují stejné položky, jinak vrací hodnota_ne. Položky jsou určeny rozdělením každého seznamu pomocí příslušného znaku oddělovače (oddělovač1 nebo oddělovač2). Pořadí položek v seznamu není relevantní. Porovnání je bez rozlišení malých a velkých písmen.

list_intersection(seznam1, seznam2, oddělovač)

class calibre.utils.formatter_functions.**BuiltinListIntersection**

list_intersection(seznam1, seznam2, oddělovač) – vrací seznam vytvořený odebrání ze seznam1 všech položek nenalezených v seznam2 pomocí porovnání bez rozlišení malých a velkých písmen. Položky v seznam1 a seznam2 jsou odděleny oddělovačem, stejně jako položky ve vráceném seznamu.

list_re(seznam_zdrojů, oddělovač, zahrnout_vrácené, volitelně_nahradit)

class calibre.utils.formatter_functions.**BuiltinListRe**

list_re(seznam_zdrojů, oddělovač, zahrnout_vrácené, volitelně_nahradit) – Vytváří seznam nejdříve rozdělením seznam_zdrojů na položky pomocí znaku oddělovače. Pro každou položku na seznamu kontroluje, zda odpovídá zahrnout_vrácené. Pokud ano, pak ji přidá na seznam vrácených. Pokud volitelně_nahradit není prázdný řetězec, pak použije náhradu před přidáním položky na seznam vrácených.

list_re_group(src_list, separator, include_re, search_re, group_1_template, ...)

class calibre.utils.formatter_functions.**BuiltinListReGroup**

list_re_group(src_list, separator, include_re, search_re, group_1_template, ...) – Jako list_re kromě toho, že nahrazení nejsou volitelná. Používáre_group(list_item, search_re, group_1_template, ...) při provádění nahrazení na seznamu výsledků.

list_sort(seznam, směr, oddělovač)

class calibre.utils.formatter_functions.**BuiltinListSort**

list_sort(seznam, směr, oddělovač) – vrací seznam seřazený pomocí řazení bez rozlišení malých a velkých písmen. Pokud je směr nula, je seznam seřazen vzestupně, jinak sestupně. Položky seznamu jsou odděleny oddělovačem, stejně jako položky ve vráceném seznamu.

list_union(seznam1, seznam2, oddělovač)

class calibre.utils.formatter_functions.**BuiltinListUnion**

list_union(seznam1, seznam2, oddělovač) – vrací seznam vytvořený sloučením položek v seznam1 a seznam2, odebráním duplicitních položek pomocí porovnání bez rozlišení malých a velkých písmen. Pokud se položky liší velikostí písmen, je použita ta z seznam1. Položky v seznam1 a seznam2 jsou odděleny oddělovačem, stejně jako položky ve vráceném seznamu.

subitems(hodnota, počáteční_index, konečný_index)

class calibre.utils.formatter_functions.**BuiltinSubitems**

subitems(hodnota, počáteční_index, konečný_index) – tato funkce je používána na rozdělení seznamů položek, jako jsou žánry. Interpretuje hodnotu jako čárkami oddělený seznam položek, kde každá položka je tečkami oddělený seznam. Vrací nový seznam vytvořený nejdříve nalezením všech tečkami oddělených položek, pak pro každou takovou položku extrahováním komponent od *počáteční_index* po *konečný_index* a pak zkombinováním výsledků zpět dohromady. První komponenta v tečkami odděleném seznamu má index nula. Pokud je index záporný, pak se počítá od konce seznamu. Jako speciální příklad je koncový_index nula považován za délku seznamu. Příklady používající režim základní šablony a předpokládající, že hodnota #genre "A.B.C": {#genre:subitems(0,1)} vrací "A". {#genre:subitems(0,2)} vrací "A.B". {#genre:subitems(1,0)} vrací "B.C". Před-

pokládající, že hodnota #genre "A.B.C, D.E.F", {#genre:subitems(0,1)} vrací "A, D". {#genre:subitems(0,2)} vrací "A.B, D.E"

sublist(hodnota, počáteční_index, konečný_index, oddělovač)

class calibre.utils.formatter_functions.**BuiltinSublist**

sublist(hodnota, počáteční_index, konečný_index, oddělovač) – interpretuje hodnotu jako seznam položek oddělený *oddělovačem*, vrací nový seznam vytvořený od položky *počáteční_index* po *konečný_index*. První položka je číslo nula. Pokud je index záporný, pak se počítá od konce seznamu. Jako speciální příklad je koncový_index nula považován za délku seznamu. Příklady používající režim základní šablony a předpokládající, že sloupec štítky (který je čárkami oddělený) obsahující "A, B, C": {tags:sublist(0,1,,)} vrací "A". {tags:sublist(-1,0,,)} vrací "C". {tags:sublist(0,-1,,)} vrací "A, B".

Other

assign(id, val)

class calibre.utils.formatter_functions.**BuiltinAssign**

assign(id, val) – přiřadí val k id, pak vrací val. id musí být identifikátor, ne výraz

print(a, b, ...)

class calibre.utils.formatter_functions.**BuiltinPrint**

print(a, b, ...) – vypíše parametry do standardního výstupu. Pokud nespustíte Calibre z příkazové řádky (calibre-debug -g), výstup půjde do černé díry.

Recursion

eval(template)

class calibre.utils.formatter_functions.**BuiltinEval**

eval(template) – vyhodnocuje šablonu, předává místní proměnné (ty 'přiřazené') namísto metadat knihy. Toto umožňuje pomocí zpracování šablony vytvořit složité výsledky z místních proměnných. Protože znaky { a } jsou speciální, musíte použít [[pro znak { a]] pro znak }; jsou převedeny automaticky. Pamatujte také na to, že předpony a přípony (syntaxe |*prefix*|*suffix*) nemohou být použity v parametru této funkce při použití režimu šablony programu.

template(x)

class calibre.utils.formatter_functions.**BuiltinTemplate**

template(x) – vyhodnocuje x jako šablonu.Vyhodnocení se provádí ve vlastním kontextu, což znamená, že proměnné nejsou sdíleny mezi volajícím a vyhodnocením šablony. Protože znaky { a } jsou speciální, musíte použít [[pro znak { a]] pro znak }; jsou převedeny automaticky. Například template('[[title_sort]]') vyhodnotí šablonu {title_sort} a vrátí její hodnotu. Pamatujte také na to, že předpony a přípony (syntaxe |*prefix*|*suffix*) nemohou být použity v parametru této funkce při použití režimu šablony programu.

Relational

cmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinCmp**

cmp(x, y, lt, eq, gt) – porovnává x a y po převedení obou na čísla. Vrací lt, když x < y. Vrací eq, když x == y. Jinak vrací gt.

first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result)

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

first_matching_cmp(val, cmp1, result1, cmp2, r2, ..., else_result) – porovnává "val < cmpN" v posloupnosti, vrací resultN pro první porovnání, které uspěje. Vrací else_result, když žádné porovnání neuspěje. Příklad: first_matching_cmp(10,5,"malý",10,"střední",15,"velký","obrovský") vrací "velký". Stejný příklad s první hodnotou 16 vrací "obrovský".

strcmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmp**

strcmp(x, y, lt, eq, gt) – dělá porovnání x a y jako řetězců bez rozlišení malých a velkých písmen. Vrací lt, když x < y. Vrací eq, když x == y. Jinak vrací gt.

String case changes

capitalize(hodnota)

class calibre.utils.formatter_functions.**BuiltinCapitalize**

capitalize(hodnota) – vrací hodnotu pole převedenou na velká počáteční písmena

lowercase(hodnota)

class calibre.utils.formatter_functions.**BuiltinLowercase**

lowercase(hodnota) – vrací hodnotu pole malými písmeny

titlecase(hodnota)

class calibre.utils.formatter_functions.**BuiltinTitlecase**

titlecase(hodnota) – vrací hodnotu pole se všemi prvními písmeny velkými

uppercase(hodnota)

class calibre.utils.formatter_functions.**BuiltinUppercase**

uppercase(hodnota) – vrací hodnotu pole velkými písmeny

String manipulation

re(hodnota, vzor, náhrada)

class calibre.utils.formatter_functions.**BuiltinRe**

re(hodnota, vzor, náhrada) – vrací pole po použití regulárního výrazu. Všechny instance *vzoru* jsou nahrazeny *náhradou*. Jako v celém Calibre jsou toto regulární výrazy kompatibilní s Pythonem

re_group(val, pattern, template_for_group_1, for_group_2, ...)

class calibre.utils.formatter_functions.**BuiltinReGroup**

re_group(val, pattern, template_for_group_1, for_group_2, ...) – vrací řetězec vytvořený použitím vzoru regulárního výrazu (pattern) na hodnotu (val) a nahrazením každé odpovídající instance řetězcem vypočítaným nahrazením každé odpovídající skupiny hodnotou vrácenou odpovídající šablonou (tempůate_for_group_1). Původní odpovídající hodnota pro skupinu je dostupná jako $. V režimu programu šablony, jako pro šablonu a funkce eval, použijte [[pro { a]] pro }. Následující příklad v režimu programu šablony hledá série s více než jedním slovem a prvním slovem velkými písmeny: {series:'re_group($, "(S*)(.*)", "[[$:uppercase()]]", "[[$]]")'}

shorten(hodnota, znaky vlevo, text uprostřed, znaky vpravo)

class calibre.utils.formatter_functions.**BuiltinShorten**

shorten(hodnota, znaky vlevo, text uprostřed, znaky vpravo) – vrací zkrácenou verzi pole, skládající se ze *znaky vlevo* znaků od začátku pole, následované *text uptostřed*, následovný *znaky vpravo* znaků od konce řetězce. *Znaky vlevo* a *znaky vpravo* musí být celá čísla. Například předpokládejme, že název knihy je *Život a doba soudce Roye Beana* a chcete, aby se vešel do míst s nanejvýše 15 znaky. Pokud použijete {title:shorten(9,-,5)}, výsledek bude *Život a d-eana*. Pokud je délka pole menší než *znaky vlevo* + *znaky vpravo* + délka *text uprostřed*, pak bude pole použito nezměněno. Například název *Kopule* by nebyl změněn.

strcat(a, b, ...)

class calibre.utils.formatter_functions.**BuiltinStrcat**

strcat(a, b, ...) – může brát jakýkoliv počet parametrů. Vrací řetězec vytvořený spojením všech parametrů

strcat_max(max, řetězec1, předpona2, řetězec2, ...)

class calibre.utils.formatter_functions.**BuiltinStrcatMax**

strcat_max(max, řetězec1, předpona2, řetězec2, ...) – Vrací řetězec vytvořený spojením parametrů. Vrácená hodnota je spuštěna na řetězec1. Páry *předpona, řetězec* jsou přidávány na konec hodnoty, dokud je výsledný řetězec kratší než *max*. Řetězec1 je vrácen, dokonce i když je řetězec1 delší než max. Můžete předat tolik párů *předpona, řetězec*, kolik si přejete.

strlen(a)

class calibre.utils.formatter_functions.**BuiltinStrlen**

strlen(a) – vrací délku řetězce předanou jako parametr

substr(řetězec, počátek, konec)

class calibre.utils.formatter_functions.**BuiltinSubstr**

substr(řetězec, počátek, konec) – vrací počáteční až koncové znaky řetězce. První znak v řetězci je nultý znak. Pokud je konec záporný, pak označuje, kolik znaků počítat zprava. Pokud je konec nula, pak označuje poslední znak. Například substr('12345', 1, 0) vrací '2345' a substr('12345', 1, -1) vrací '234'.

swap_around_comma(hodnota)

class calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

swap_around_comma(hodnota) – zadává se hodnota v podobě "B, A", vrací "A B". Toto je nejužitečnější pro převod jmen z formátu "příjmení, jméno" na "jméno příjmení". Pokud zde není čárka, funkce vrací hodnotu nezměněnou.

transliterate(a)

class calibre.utils.formatter_functions.**BuiltinTransliterate**

transliterate(a) – Vrací řetězec v latince vytvořený přibližným odhadem zvuku slov ve zdrojovém řetězci. Například pokud je zdroj "Фёдор Михайлович Достоевский", funkce vrací "Fiodor Mikhailovich Dostoievskii".

API of the Metadata objects The python implementation of the template functions is passed in a Metadata object. Knowing it's API is useful if you want to define your own template functions.

class calibre.ebooks.metadata.book.base.**Metadata**(*title*, *authors=(u'Neznxe1mxfd'*, *)*, *other=None*, *template_cache=None*, *formatter=None*)

A class representing all the metadata for a book. The various standard metadata fields are available as attributes of this object. You can also stick arbitrary attributes onto this object.

Metadata from custom columns should be accessed via the get() method, passing in the lookup name for the column, for example: "#mytags".

Use the *is_null()* (stránka 156) method to test if a field is null.

This object also has functions to format fields into strings.

The list of standard metadata fields grows with time is in *STANDARD_METADATA_FIELDS* (stránka 157).

Please keep the method based API of this class to a minimum. Every method becomes a reserved field name.

is_null(*field*)
Return True if the value of field is null in this object. 'null' means it is unknown or evaluates to False. So a title of _('Unknown') is null or a language of 'und' is null.

Be careful with numeric fields since this will return True for zero as well as None.

Also returns True if the field does not exist.

deepcopy(*class_generator=<function <lambda>>*)
Do not use this method unless you know what you are doing, if you want to create a simple clone of this object, use `deepcopy_metadata()` instead. Class_generator must be a function that returns an instance of Metadata or a subclass of it.

get_identifiers()
Return a copy of the identifiers dictionary. The dict is small, and the penalty for using a reference where a copy is needed is large. Also, we don't want any manipulations of the returned dict to show up in the book.

set_identifiers(*identifiers*)
Set all identifiers. Note that if you previously set ISBN, calling this method will delete it.

set_identifier(*typ*, *val*)
If val is empty, deletes identifier of type typ

standard_field_keys()
return a list of all possible keys, even if this book doesn't have them

custom_field_keys()
return a list of the custom fields in this book

all_field_keys()
All field keys known by this instance, even if their value is None

metadata_for_field(*key*)
return metadata describing a standard or custom field.

all_non_none_fields()
Return a dictionary containing all non-None metadata fields, including the custom ones.

get_standard_metadata(*field*, *make_copy*)
return field metadata from the field if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

get_all_standard_metadata(*make_copy*)
return a dict containing all the standard field metadata associated with the book.

get_all_user_metadata(*make_copy*)
return a dict containing all the custom field metadata associated with the book.

get_user_metadata(*field*, *make_copy*)
return field metadata from the object if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

set_all_user_metadata(*metadata*)
store custom field metadata into the object. Field is the key name not the label

set_user_metadata(*field*, *metadata*)
store custom field metadata for one column into the object. Field is the key name not the label

template_to_attribute(*other*, *ops*)
Takes a list [(src,dest), (src,dest)], evaluates the template in the context of other, then copies the result to self[dest]. This is on a best-efforts basis. Some assignments can make no sense.

smart_update(*other*, *replace_metadata=False*)
Merge the information in *other* into self. In case of conflicts, the information in *other* takes precedence, unless the information in *other* is NULL.

format_field(*key*, *series_with_index=True*)
Returns the tuple (display_name, formatted_value)

to_html()
A HTML representation of this object.

calibre.ebooks.metadata.book.base.**STANDARD_METADATA_FIELDS**
The set of standard metadata fields.

```
'''
All fields must have a NULL value represented as None for simple types,
an empty list/dictionary for complex types and (None, None) for cover_data
'''

SOCIAL_METADATA_FIELDS = frozenset([
    'tags',             # Ordered list
    'rating',           # A floating point number between 0 and 10
    'comments',         # A simple HTML enabled string
    'series',           # A simple string
    'series_index',     # A floating point number
    # Of the form { scheme1:value1, scheme2:value2}
    # For example: {'isbn':'123456789', 'doi':'xxxx', ... }
    'identifiers',
])

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset([
    'isbn',
])

PUBLICATION_METADATA_FIELDS = frozenset([
    'title',            # title must never be None. Should be _('Unknown')
    # Pseudo field that can be set, but if not set is auto generated
    # from title and languages
    'title_sort',
    'authors',          # Ordered list. Must never be None, can be [_('Unknown')]
    'author_sort_map',  # Map of sort strings for each author
    # Pseudo field that can be set, but if not set is auto generated
    # from authors and languages
    'author_sort',
    'book_producer',
    'timestamp',        # Dates and times must be timezone aware
    'pubdate',
```

```
    'last_modified',
    'rights',
    # So far only known publication type is periodical:calibre
    # If None, means book
    'publication_type',
    'uuid',             # A UUID usually of type 4
    'languages',        # ordered list of languages in this publication
    'publisher',        # Simple string, no special semantics
    # Absolute path to image file encoded in filesystem_encoding
    'cover',
    # Of the form (format, data) where format is, for e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
    ])

BOOK_STRUCTURE_FIELDS = frozenset([
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
    ])

USER_METADATA_FIELDS = frozenset([
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
])

DEVICE_METADATA_FIELDS = frozenset([
    'device_collections',   # Ordered list of strings
    'lpath',                # Unicode, / separated
    'size',                 # In bytes
    'mime',                 # Mimetype of the book file being represented

])

CALIBRE_METADATA_FIELDS = frozenset([
    'application_id',   # An application id, currently set to the db_id.
    'db_id',            # the calibre primary key of the item.
    'formats',          # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of author to an associated hyperlink
    'author_link_map',

    ]
)

ALL_METADATA_FIELDS =       SOCIAL_METADATA_FIELDS.union(
                            PUBLICATION_METADATA_FIELDS).union(
                            BOOK_STRUCTURE_FIELDS).union(
                            USER_METADATA_FIELDS).union(
                            DEVICE_METADATA_FIELDS).union(
                            CALIBRE_METADATA_FIELDS)

# All fields except custom fields
```

```
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS)

# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED =     frozenset(['title', 'title_sort', 'authors',
                                      'author_sort', 'author_sort_map',
                                      'cover_data', 'tags', 'languages',
                                      'identifiers'])

# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL =  frozenset(['lpath', 'size', 'comments', 'thumbnail'])

# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS =       SOCIAL_METADATA_FIELDS.union(
                           PUBLICATION_METADATA_FIELDS).union(
                           BOOK_STRUCTURE_FIELDS).union(
                           DEVICE_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS) - \
                           SC_FIELDS_NOT_COPIED.union(
                           SC_FIELDS_COPY_NOT_NULL)

SERIALIZABLE_FIELDS =      SOCIAL_METADATA_FIELDS.union(
                           USER_METADATA_FIELDS).union(
                           PUBLICATION_METADATA_FIELDS).union(
                           CALIBRE_METADATA_FIELDS).union(
                           DEVICE_METADATA_FIELDS) - \
                           frozenset(['device_collections', 'formats',
                               'cover_data'])
                           # these are rebuilt when needed
```

Using general program mode

For more complicated template programs, it is sometimes easier to avoid template syntax (all the *{* and *}* characters), instead writing a more classical-looking program. You can do this in calibre by beginning the template with *program:*. In this case, no template processing is done. The special variable *$* is not set. It is up to your program to produce the correct results.

One advantage of *program:* mode is that the brackets are no longer special. For example, it is not necessary to use *[[* and *]]* when using the *template()* function. Another advantage is that program mode templates are compiled to Python and can run much faster than templates in the other two modes. Speed improvement depends on the complexity of the templates; the more complicated the template the more the improvement. Compilation is turned off or on using the tweak `compile_gpm_templates` (Compile General Program Mode templates to Python). The main reason to turn off compilation is if a compiled template does not work, in which case please file a bug report.

The following example is a *program:* mode implementation of a recipe on the MobileRead forum: "Put series into the title, using either initials or a shortened form. Strip leading articles from the series name (any)." For example, for the book The Two Towers in the Lord of the Rings series, the recipe gives *LotR [02] The Two Towers*. Using standard templates, the recipe requires three custom columns and a plugboard, as explained in the following:

The solution requires creating three composite columns. The first column is used to remove the leading articles. The second is used to compute the 'shorten' form. The third is to compute the 'initials' form. Once you have these columns, the plugboard selects between them. You can hide any or all of the three columns on the library view:

```
First column:
Name: #stripped_series.
Template: {series:re(^(A|The|An)\s+,)||}

Second column (the shortened form):
Name: #shortened.
Template: {#stripped_series:shorten(4,-,4)}

Third column (the initials form):
Name: #initials.
Template: {#stripped_series:re(([^\s])[^\s]+(\s|$),\1)}

Plugboard expression:
Template:{#stripped_series:lookup(.\s,#initials,.,#shortened,series)}{series_index:0>2.0f| [|] }{tit
Destination field: title

This set of fields and plugboard produces:
Series: The Lord of the Rings
Series index: 2
Title: The Two Towers
Output: LotR [02] The Two Towers

Series: Dahak
Series index: 1
Title: Mutineers Moon
Output: Dahak [01] Mutineers Moon

Series: Berserkers
Series Index: 4
Title: Berserker Throne
Output: Bers-kers [04] Berserker Throne

Series: Meg Langslow Mysteries
Series Index: 3
Title: Revenge of the Wrought-Iron Flamingos
Output: MLM [03] Revenge of the Wrought-Iron Flamingos
```

The following program produces the same results as the original recipe, using only one custom column to hold the results of a program that computes the special title value:

```
Custom column:
Name: #special_title
Template: (the following with all leading spaces removed)
    program:
    #       compute the equivalent of the composite fields and store them in local variables
        stripped = re(field('series'), '^(A|The|An)\s+', '');
        shortened = shorten(stripped, 4, '-' ,4);
        initials = re(stripped, '[^\w]*(\w?)[^\s]+(\s|$)', '\1');

    #       Format the series index. Ends up as empty if there is no series index.
    #       Note that leading and trailing spaces will be removed by the formatter,
    #       so we cannot add them here. We will do that in the strcat below.
    #       Also note that because we are in 'program' mode, we can freely use
    #       curly brackets in strings, something we cannot do in template mode.
        s_index = template('{series_index:0>2.0f}');

    #       print(stripped, shortened, initials, s_index);
```

```
    #       Now concatenate all the bits together. The switch picks between
    #       initials and shortened, depending on whether there is a space
    #       in stripped. We then add the brackets around s_index if it is
    #       not empty. Finally, add the title. As this is the last function in
    #       the program, its value will be returned.
        strcat(
            switch( stripped,
                    '.\s', initials,
                    '.', shortened,
                    field('series')),
            test(s_index, strcat(' [', s_index, '] '), ''),
            field('title'));

Plugboard expression:
Template:{#special_title}
Destination field: title
```

It would be possible to do the above with no custom columns by putting the program into the template box of the plugboard. However, to do so, all comments must be removed because the plugboard text box does not support multi-line editing. It is debatable whether the gain of not having the custom column is worth the vast increase in difficulty caused by the program being one giant line.

User-defined Template Functions

You can add your own functions to the template processor. Such functions are written in python, and can be used in any of the three template programming modes. The functions are added by going to Preferences -> Advanced -> Template Functions. Instructions are shown in that dialog.

Special notes for save/send templates

Special processing is applied when a template is used in a *save to disk* or *send to device* template. The values of the fields are cleaned, replacing characters that are special to file systems with underscores, including slashes. This means that field text cannot be used to create folders. However, slashes are not changed in prefix or suffix strings, so slashes in these strings will cause folders to be created. Because of this, you can create variable-depth folder structure.

For example, assume we want the folder structure *series/series_index - title*, with the caveat that if series does not exist, then the title should be in the top folder. The template to do this is:

```
{series:||/}{series_index:|| - }{title}
```

The slash and the hyphen appear only if series is not empty.

The lookup function lets us do even fancier processing. For example, assume that if a book has a series, then we want the folder structure *series/series index - title.fmt*. If the book does not have a series, then we want the folder structure *genre/author_sort/title.fmt*. If the book has no genre, we want to use 'Unknown'. We want two completely different paths, depending on the value of series.

To accomplish this, we:

1. Create a composite field (call it AA) containing `{series}/{series_index} - {title'}`. If the series is not empty, then this template will produce *series/series_index - title*.
2. Create a composite field (call it BB) containing `{#genre:ifempty(Unknown)}/{author_sort}/{title}`. This template produces *genre/author_sort/title*, where an empty genre is replaced with *Unknown*.

3. Set the save template to `{series:lookup(.,AA,BB)}`. This template chooses composite field AA if series is not empty, and composite field BB if series is empty. We therefore have two completely different save paths, depending on whether or not *series* is empty.

Templates and Plugboards

Plugboards are used for changing the metadata written into books during send-to-device and save-to-disk operations. A plugboard permits you to specify a template to provide the data to write into the book's metadata. You can use plugboards to modify the following fields: authors, author_sort, language, publisher, tags, title, title_sort. This feature helps people who want to use different metadata in books on devices to solve sorting or display issues.

When you create a plugboard, you specify the format and device for which the plugboard is to be used. A special device is provided, save_to_disk, that is used when saving formats (as opposed to sending them to a device). Once you have chosen the format and device, you choose the metadata fields to change, providing templates to supply the new values. These templates are *connected* to their destination fields, hence the name *plugboards*. You can, of course, use composite columns in these templates.

When a plugboard might apply (content server, save to disk, or send to device), calibre searches the defined plugboards to choose the correct one for the given format and device. For example, to find the appropriate plugboard for an EPUB book being sent to an ANDROID device, calibre searches the plugboards using the following search order:

- a plugboard with an exact match on format and device, e.g., `EPUB` and `ANDROID`
- a plugboard with an exact match on format and the special `any device` choice, e.g., `EPUB` and `any device`
- a plugboard with the special `any format` choice and an exact match on device, e.g., `any format` and `ANDROID`
- a plugboard with `any format` and `any device`

The tags and authors fields have special treatment, because both of these fields can hold more than one item. A book can have many tags and many authors. When you specify that one of these two fields is to be changed, the template's result is examined to see if more than one item is there. For tags, the result is cut apart wherever calibre finds a comma. For example, if the template produces the value `Thriller, Horror`, then the result will be two tags, `Thriller` and `Horror`. There is no way to put a comma in the middle of a tag.

The same thing happens for authors, but using a different character for the cut, a *&* (ampersand) instead of a comma. For example, if the template produces the value `Blogs, Joe&Posts, Susan`, then the book will end up with two authors, `Blogs, Joe` and `Posts, Susan`. If the template produces the value `Blogs, Joe;Posts, Susan`, then the book will have one author with a rather strange name.

Plugboards affect the metadata written into the book when it is saved to disk or written to the device. Plugboards do not affect the metadata used by `save to disk` and `send to device` to create the file names. Instead, file names are constructed using the templates entered on the appropriate preferences window.

Helpful Tips

You might find the following tips useful.

- Create a custom composite column to test templates. Once you have the column, you can change its template simply by double-clicking on the column. Hide the column when you are not testing.
- Templates can use other templates by referencing a composite custom column.
- In a plugboard, you can set a field to empty (or whatever is equivalent to empty) by using the special template `{}`. This template will always evaluate to an empty string.
- The technique described above to show numbers even if they have a zero value works with the standard field series_index.

1.9.4 Vše o používání regulárních výrazů v Calibre

Regulární výrazy jsou funkce používané na mnoha místech v Calibre pro provádění propracovaného zpracování obsahu e-knihy a metadat. Tento kurz je mírný úvod do problematiky používání regulárních výrazů v Calibre.

Obsah

- *Nejdříve malé varování a malé povzbuzení* (stránka 163)
- *Kde v Calibre můžete použít regulární výrazy?* (stránka 163)
- *Co to u všech všudy* je *regulární příkaz?* (stránka 163)
- *Zajímá vás vysvětlení?* (stránka 164)
- *To nezní tak špatně. Co dál?* (stránka 164)
- *Parádní! Začíná to dávat smysl!* (stránka 164)
- *No, tyto speciální znaky jsou velice elegantní a tak, ale co když chci porovnat tečku nebo otazník?* (stránka 165)
- *Takže, jaké jsou nejužitečnější sady?* (stránka 165)
- *Ale když budu mít několik různých řetězců, které chci porovnat, zkomplikuje se to?* (stránka 165)
- *Zapomněl jsi...* (stránka 166)
- *Na začátku jsi říkal, že existuje způsob, aby regulární výraz nerozlišoval malá a velká písmena?* (stránka 166)
- *Myslím, že teď už začínám ty regulární výrazy chápat... Jak je použiju v Calibre?* (stránka 166)
  - *Převody* (stránka 166)
  - *Přidávání knih* (stránka 167)
  - *Hromadná úprava metadat* (stránka 167)
- *Poděkování* (stránka 167)

Nejdříve malé varování a malé povzbuzení

Toto bude nevyhnutelně poněkud technické – ostatně regulární výrazy jsou technickým nástrojem pro to technické záležitosti. Budu muset použít nějaký žargon a pojmy, které se mohou zdát složité a spletité. Budu se snažit vysvětlit tyto pojmy co nejjasněji, ale opravdu není možné obejít se bez jejich použití. Jak již bylo řečeno, nenechte se odradit žádným žargonem, protože se snažím vysvětlit všechno nové. A zatímco samotné regulární výrazy vám můžou připadat jako tajemná, černé magie (nebo, abychom byli prozaičtější, náhodný řetězec nesmyslných písmen a znaků), slibuji, že nejsou vůbec tak složité. Dokonce i ti, kteří pochopili regulární výrazy opravdu dobře, mají problémy se čtením těch složitějších, ale jejich psaní není tak obtížné – vytvářítе výraz krok za krokem. Takže vykročte a následujte mě do králičí nory.

Kde v Calibre můžete použít regulární výrazy?

Existuje několik míst, kde Calibre používá regulární výrazy. Je zde Najít a nahradit ve volbách převodu, detekce metadat z názvů souborů v nastavení importu a Najít a nahradit při hromadné úpravě metadat knih. Editor knih Calibre může také použít regulární výrazy ve své funkci Najít a nahradit.

Co to u všech všudy *je* regulární příkaz?

Regulární výraz je způsob, jak popsat sady řetězců. Jeden regulární výraz může *odpovídat* velkému počtu různých řetězců. Proto jsou regulární výrazy tak mocné – jsou stručným způsobem popisu potenciálně velkého počtu variací.

Poznámka: Řetězec zde používám ve smyslu, v jakém se používá v programovacích jazycích: řetězec jednoho nebo více znaků, znaků včetně skutečných znaků, čísel, interpunkčních znamének a takzvaných prázdných znaků (konce řádků, tabulátorů atd). Pamatujte, že velká a malá písmena nejsou obecně považována za stejná, tedy "a" je jiný

znak než “A” a tak dále. V Calibre regulární výrazy rozlišují velikost písmen v panelu hledání, ale ne ve volbách převodu. Existuje způsob, aby každý regulární výraz rozlišoval velikost písmen, ale to probereme později. Trochu se to komplikuje, protože regulární výrazy umožňují variace v řetězcích, se kterými se porovnávají, takže jeden výraz může odpovídat více řetězcům, což je důvod, proč se je lidé vůbec obtěžují používat. Více si o tom povíme za chvíli.

Zajímá vás vysvětlení?

No, proto jsme tady. Za prvé, toto je nejdůležitější koncept v regulárních výrazech: *Řetězec sám o sobě je regulární výraz, který odpovídá sobě samému*. To znamená, že když chci pomocí regulárního výrazu porovnat řetězec `"Hello, World!"`, použitý regulární výraz by byl `Hello, World!`. Ano, opravdu je to tak jednoduché. Ale všimněte si, že toto odpovídá *pouze* přesnému řetězci `"Hello, World!"`, ne například `"Hello, wOrld!"` nebo `"hello, world!"` nebo jakékoliv jiné takové variaci.

To nezní tak špatně. Co dál?

Dál začínají opravdu dobré věci. Pamatujete si, jak jsem říkal, že regulární výrazy mohou odpovídat více řetězcům? Tady to bude trochu složitější. Řekněme, jako poněkud praktičtější cvičení, že e-kniha, kterou chcete převést, má ošklivé zápatí počítající stránky, jako “Strana 5 z 423”. Je zřejmé, že číslo stránky bude růst od 1 do 423, takže budete muset porovnat 423 různých řetězců, že? Špatně, ve skutečnosti: regulární výrazy vám umožňují definovat sady znaků, které jsou porovnávány. Abyste definovali sadu, dejte všechny znaky, které chcete mít v sadě, do hranatých závorek. Takže například sada `[abc]` by odpovídala buď znaku “a”, “b” nebo “c”. *Sady budou vždy porovnávat pouze jeden ze znaků v sadě*. “Rozumějí” rozsahům znaků, takže pokud chcete porovnat všechna malá písmena, použili byste sadu `[a-z]`, pro malá a velká písmena byste použili `[a-zA-Z]` a tak dále. Chápete to? Takže samozřejmě pomocí výrazu `Strana [0-9] z 423` byste byli schopni porovnat prvních 9 stránek, čímž se sníží počet potřebných výrazů na tři: Druhý výraz `Strana [0-9][0-9] z 423` by porovnal všechna dvojciferná čísla stránek, a jsem si jistý, že umíte odhadnout, jak by vypadal třetí výraz. Ano, do toho. Napište ho.

Parádní! Začíná to dávat smysl!

Doufal jsem, že to řeknete. Ale připravte se, teď to bude ještě lepší! Právě jsme viděli, že pomocí sad jsme mohli porovnat jeden z několika znaků najednou. Ale můžete dokonce opakovat znak nebo sadu, čímž snížíte počet výrazů potřebných pro zpracování výše uvedeného příkladu s čísly stránek na jeden. Ano, JEDEN! Jste nadšení? Měli byste být! Funguje to takhle: Některé takzvané speciální znaky, “+”, ”?” a “*”, *opakují jednotlivý předcházející prvek*. (Prvek znamená buď jeden znak, sadu znaků, řídící sekvenci nebo skupinu (o těch posledních dvou se budeme učit později.) – zkrátka každý jednotlivý prvek v regulárním výrazu) Tyto znaky se nazývají zástupné znaky nebo kvantifikátory. Abychom byli přesnější, ”?” odpovídá *0 nebo 1* výskytů předcházejícího prvku, “*” odpovídá *0 nebo více* výskytů předcházejícího prvku a “+” odpovídá *1 nebo více* výskytů předcházejícího prvku. Několik příkladů: Výraz `a?` by odpovídal buď “” (což je prázdný řetězec, není nezbytně užitečný v tomto případě), nebo “a”, výraz `a*` by odpovídal “”, “a”, “aa” nebo libovolnému počtu a v řadě, a konečně výraz `a+` by odpovídal “a”, “aa” nebo libovolnému počtu a v řadě (Poznámka: neodpovídal by prázdnému řetězci!). To samé pro sady: Výraz `[0-9]+` by odpovídal *každému existujícímu celému číslu*! Vím, co si myslíte, a máte pravdu: Když toto použijete ve výše uvedeném případu porovnávání čísel stránek, nebyl by to jediný výraz pro porovnání všech čísel stránek? Ano, výraz `Strana [0-9]+ z 423` by porovnal každé číslo stránky v této knize!

Poznámka: Poznámka o těchto kvantifikátorech: Obvykle se snaží porovnat tolik textu, kolik je možné, takže buďte opatrní při jejich použití. Nazývá se to “chamtivý chování” – jsem si jistý, že chápete proč. Je to komplikovanější, když, řekněme, chcete porovnat značku. Vezměme například řetězec `"<p class="calibre2">Tady je nadpis</p>"` a řekněme, že byste chtěli porovnat otevírací značku (část mezi první dvojicí hranatých závorek, značky probereme trochu více později). Mysleli byste si, že by této značce odpovídal výraz `<p.*>`, ale ve skutečnosti odpovídá celému řetězci! (Znak ”.” je další speciální znak. Odpovídá čemukoliv *kromě* konců řádků, takže v podstatě výraz `.*` by odpovídal každičkému řádku, který si dokážete

představit.) Místo toho zkuste použít ``<p.*?>``, čímž přestane být kvantifikátor "*" chamtivým. Tento výraz bude odpovídat pouze první otevírací značce, jak bylo zamýšleno. Existuje vlastně další způsob, jak toho dosáhnout: Výraz <p[^>]*> bude odpovídat té samé otevírací značce – po další části uvidíte proč. Jen vezměte na vědomí, že dost často existuje více než jeden způsob, jak napsat regulární výraz.

No, tyto speciální znaky jsou velice elegantní a tak, ale co když chci porovnat tečku nebo otazník?

Samozřejmě to můžete udělat: Stačí před jakýkoliv speciální znak vložit zpětné lomítko a je interpretován jako prostý znak bez jakéhokoliv speciálního významu. Tento pár zpětného lomítka následovaného jedním znakem se nazývá řídící sekvence, a vložení zpětného lomítka před speciální znak se nazývá uvozování toho znaku. Řídící sekvence je interpretována jako jeden prvek. Existují samozřejmě řídící sekvence, které dělají více než jen uvozování speciálních znaků, například "\t" znamená tabulátor. K některým řídícím sekvencím se dostaneme později. Jo a mimochodem, pokud jde o tyto speciální znaky: Uvažujte o kterémkoliv znaku, o kterém se budeme v tomto úvodu bavit, že má nějakou funkci, která je speciální, a tedy musí být uvozen, pokud chcete prostý znak.

Takže, jaké jsou nejužitečnější sady?

Věděl jsem, že se zeptáte. Některé užitečné sady jsou [0-9] odpovídající jednomu číslu, [a-z] odpovídající jednomu malému písmenu, [A-Z] odpovídající jednomu velkému písmenu, [a-zA-Z] odpovídající jednomu písmenu a [a-zA-Z0-9] odpovídající jednomu písmenu nebo číslu. Můžete také použít řídící sekvenci jako zjednodušení:

```
\d is equivalent to [0-9]
\w is equivalent to [a-zA-Z0-9_]
\s is equivalent to any whitespace
```

Poznámka: "Prázdný znak" je termín pro cokoliv, co nebude vytištěno. Tyto znaky zahrnují mezeru, tabulátor, posun řádku, posun strany a návrat na začátek řádku.

Jako poslední poznámka o sadách, sadu můžete také definovat jako jakýkoliv znak *kromě* těch v sadě. Uděláte to zahrnutím znaku "^" jako *úplně prvního znaku v sadě*. Takže [^a] by odpovídalo libovolnému znaku kromě "a". Tomu se říká doplňující sada. Zkratky řídící sekvence, které jsme viděli dříve, mohou být také doplněny: "\D" znamená jakýkoli nečíselný znak, takže je ekvivalentem [^0-9]. Jak už jste asi uhodli, ostatní zkratky mohou být doplněny pomocí příslušného velkého písmena namísto malého. Takže když se vrátíme k příkladu <p[^>]*> z předchozí části, můžete nyní vidět, že použitá sada znaků se pokusí porovnat jakýkoliv znak kromě uzavírací úhlové závorky.

Ale když budu mít několik různých řetězců, které chci porovnat, zkomplikuje se to?

Nebojte se, život je pořád jednoduchý. Vezměme si tento příklad: Kniha, kterou převádíte má napsán "Název" na každé liché stránce a "Autor" na každé sudé stránce. V tisku to vypadá skvěle, že jo? Ale v e-knize je to otravné. Můžete seskupit celé výrazy do normálních závorek a znak "|" vám umožní porovnat *buď* výraz s jeho pravou *nebo* levou stranou. Toto zkombinujte a máte hotovo. Je to na vás příliš? Dobře, nejdříve seskupíme výrazy pro sudé a liché stránky, takže získáme (Název)(Autor) jako naše dva potřebné výrazy. Nyní věci zjednodušíme pomocí svislé čáry ho pruhu (znak "|" se nazývá svislá čára): Pokud použijete výraz (Název|Autor), dostanete shodu buď pro "Název" (na lichých stránkách), nebo pro "Autor" (na sudých stránkách). No, nebylo to jednoduché?

Svislou čáru můžete samozřejmě použít také bez použití seskupovacích závorek. Pamatujete, když jsem říkal, že kvantifikátory opakují jim předcházející prvek? No, svislá čára funguje trochu jinak: Výraz "Název|Autor" bude také odpovídat buď řetězci "Název" nebo řetězci "Autor", stejně jako výše uvedený příklad používající seskupování. *Svislá čára vybírá mezi celým výrazem, který ji předchází a následuje*. Takže pokud byste chtěli porovnat řetězce "Calibre" a "calibre" a chtěli byste vybrat pouze mezi malým a velkým "c", museli byste použít výraz (c|C)alibre, kde seskupení zajišťuje, že bude vybráno pouze "c". Pokud byste použili c|Calibre, získali byste shodu na řetězec

“c” nebo na řetězec “Calibre”, což není to, co jsme chtěli. Stručně řečeno: Pokud si nejste jisti, použijte seskupování společně se svislou čárou.

Zapomněl jsi...

... počkejte chvilku, je tu ještě jedna poslední, opravdu užitečná věc, kterou můžete dělat se skupinami. Pokud máte skupinu, kterou jste předtím porovnal, můžete použít odkazy na tuto skupinu později ve výrazu: Skupiny jsou číslovány od 1, a odkazujete na ně uvozením čísla skupiny, na kterou chcete odkázat, tedy pátá skupina by byla odkázána `\5`. Takže pokud jste hledali `([^ ]+) \1` v řetězci “Test Test”, porovnali byste celý řetězec!

Na začátku jsi říkal, že existuje způsob, aby regulární výraz nerozlišoval malá a velká písmena?

Ano, říkal, díky za věnování pozornosti a připomenutí. Můžete říct Calibre, jak chcete řešit některé věci pomocí tzv. vlajek. Vlajky zahrnete do svého výrazu pomocí speciální konstrukce `(?sem přijdou vlajky)`, kde byste samozřejmě nahradili “sem přijdou vlajky” konkrétními vlajkami, které chcete. Pro ignorování velikosti písmen je vlajka `i`, takže zahrnete do svého výrazu `(?i)`. Takže `test(?i)` by porovnal “Test”, “tEst”, “TEst” a jakýkoliv případ variace si dokážete vymyslet.

Další užitečná vlajka umožňuje, aby tečka odpovídala všem libovolným znakům, *včetně* nového řádku, a to vlajka `s`. Pokud chcete použít ve výrazu více vlajek, stačí je dát do stejného zápisu: `(?is)` by ignorovalo velikost písmen a tečka by odpovídala všemu. Nezáleží na tom, kterou vlajkou uvedete první, `(?si)` by bylo ekvivalentem výše uvedeného. Mimochodem, dobré místo pro uvádění vlajek ve vašem výrazu je buď úplně na začátku nebo úplně na konci. Takto se nezamíchají s ničím jiným.

Myslím, že teď už začínám ty regulární výrazy chápat... Jak je použiju v Calibre?

Převody

Začněme s nastavením převodu, což je opravdu elegantní. Do části Hledat a nahradit můžete zadat regulární výraz, který popisuje řetězec, který bude při převodu nahrazen. Elegantní na tom je průvodce. Klikněte na kouzelnickou hůlku a získáte náhled toho, co Calibre “vidí” během procesu převodu. Přejděte dolů na řetězec, který chcete odebrat, vyberte ho a zkopírujte, vložte ho do pole regulárního výrazu v horní části okna. Pokud jsou zde proměnné části, jako jsou čísla stránek nebo tak, použijte pro jejich pokrytí sady a kvantifikátory, a když už jsme u toho, nezapomeňte uvodit speciální znaky, pokud tu nějaké jsou. Stiskněte tlačítko označené *Testovat* a Calibre zvýrazní části, které budou nahrazeny, pokud použijete regulární výraz. Jakmile jste spokojeni, klikněte na tlačítko OK a proveďte převod. Buďte opatrní, pokud váš zdroj převodu obsahuje značky, jako tento příklad:

```
Maybe, but the cops feel like you do, Anita. What's one more dead vampire?
New laws don't change that. </p>
<p class="calibre4"> <b class="calibre2">Generated by ABC Amber LIT Conv
<a href="http://www.processtext.com/abclit.html" class="calibre3">erter,
http://www.processtext.com/abclit.html</a></b></p>
<p class="calibre4"> It had only been two years since Addison v. Clark.
The court case gave us a revised version of what life was
```

(nestydatě vytrženo z tohoto vlákna[75]). Museli byste také odstranit některé značky. V tomto příkladu bych doporučoval začít značkou `<b class="calibre2">`, nyní budete muset skončit odpovídající uzavírací značkou (otevírací značky jsou `<značka>`, uzavírací značky jsou `</značka>`), což je v tomto případě jednoduše následující `</b>`. (Podívejte se do dobré příručky k HTML nebo se zeptejte ve fóru, pokud si v tomto bodě nejste jisti.) Počáteční značku lze popsat pomocí `<b.*?>`, uzavírací značku pomocí `</b>`, takže bychom mohli odstranit vše mezi těmito značkami pomocí `<b.*?>.*?</b>`. Ale použití tohoto výrazu by byl špatný nápad, protože odstraňuje vše, co je uzavřeno značkami <b> (který mimochodem vykreslí uzavřený text tučným písmem), a vsaďte se, že

[75] http://www.mobileread.com/forums/showthread.php?t=75594"

tímto způsobem odstraníme části knihy. Místo toho zahrňte i začátek uzavřeného řetězce, takže regulární výraz bude `<b.*?>\s*Vygeneroval\s+ABC\s+Amber\s+LIT.*?</b>`. `\s` s kvantifikátory jsou zde zahrnuty namísto výslovného použití mezer, jak je vidět v řetězci, aby byly zachyceny všechny variace řetězce, které by mohly nastat. Nezapomeňte zkontrolovat, co Calibre odstraní, když testuje nový výraz, abyste se ujistili, že neodstraníte žádné části, které chcete zachovat. Pokud zkontrolujete pouze jeden výskyt, může vám uniknout neshoda někde jinde v textu. Také pamatujte, že kdybyste omylem odstranili více nebo méně značek, než jste doopravdy chtěli, Calibre se pokusí opravit poškozený kód po provedení odstranění.

Přidávání knih

Další věc, ke které můžete použít regulární výrazy, je extrahování metadat z názvů souborů. Tuto funkci můžete najít v části nastavení "Přidávání knih". Je zde speciální funkce: Můžete použít názvy polí pro pole metadat, například `(?P<title>)` by znamenalo, že Calibre použije tuto část řetězce jako název knihy. Povolené názvy polí jsou uvedeny v oknech, společně s dalším pěkným testovacím polem. Příklad: Řekněme, že chcete importovat spoustu souborů s názvem jako je `Klasické texty: Božská komedie, Dante Alighieri.mobi`. (Samozřejmě, že tu už ve své knihovně máte, protože všichni milujeme klasickou italskou poezii) nebo `Sci-fi eposy: Trilogie Nadace, Isaac Asimov.epub`. To je samozřejmě schéma názvů, ze kterého by Calibre nevyextrahovalo žádná smysluplná data – standardní výraz pro extrahování metadat je `(?P<title>.+) - (?P,author>[^_]+)`. Regulární výraz, který by tu fungoval, by byl `[a-zA-Z]+: (?P<title>.+), (?P<author>.+)`. Pamatujte, že uvnitř skupiny pro pole metadat musíte použít výrazy k popisu toho, čemu pole skutečně odpovídá. A také pamatujte, že při použití testovacího pole, které Calibre poskytuje, musíte přidat příponu souboru k testovanému souboru, jinak nezískáte vůbec žádnou shodu, i když použijete fungující výraz.

Hromadná úprava metadat

Poslední část je hledání a nahrazování v polích metadat pomocí regulárního výrazu. K tomuto se můžete dostat výběrem více knih v knihovně a použitím hromadné úpravy metadat. Buďte velmi opatrní při používání této poslední funkce, protože může způsobit **velice špatné věci** vaší knihovně! Překontrolujte, že vaše výrazy dělají to, co chcete, pomocí testovacích polí, a označte pouze knihy, které opravdu chcete změnit! V režimu hledání regulárními výrazy můžete hledat v jednom poli, nahradit text něčím a dokonce zapsat výsledek do jiného pole. Praktický příklad: Řekněme, že vaše knihovna obsahuje knihy série Duna Franka Herberta pojmenované po vzoru `Duna 1 - Duna`, `Duna 2 - Spasitel Duny` a tak dále. Nyní chcete dostat `Duna` do pole série. Můžete to udělat hledáním `(.*?) \d+ - .*` v poli názvu a nahrazením `\1` v poli série. Vidíte, co jsem udělal? To je odkaz na první skupinu, kterou nahrazujete pole série. Nyní, když máte sérii nastavenou, potřebujete jen provést další vyhledávání `.*?` – v poli názvu a nahrazením `""` (prázdný řetězec) opět v poli názvu, a vaše metadata jsou jsou pěkně čistá a uklizená. Není to skvělé? Mimochodem, namísto nahrazení celého pole můžete také poli připojit nebo předřadit, takže pokud *chcete*, aby byla názvu knihy předřazena informace o sérii, mohli byste to udělat stejně. Jak jste si už nepochybně všimli, je zde zaškrtávací políčko označené *Rozlišovat malá a velká*, takže zde nebudete muset používat vlajky pro výběr chování.

No, je asi na čase uzavřít velice krátký úvod do regulárních výrazů. Doufám, že jsem vám ukázal dost, abyste alespoň mohli začít, a umožnil vám pokračovat v samostatném učení – dobrým výchozím bodem je Dokumentace Pythonu pro regulární výrazy[76].

Ale ještě jedna poslední varování: Regulární výrazy jsou výkonné, ale také je velice snadné je pokazit. Calibre poskytuje opravdu skvělé možnosti pro testování, abyste viděli, jestli se vaše výrazy chovat podle očekávání. Používejte je. Pokuste se nestřelit se do nohy. (Bože, tenhle výraz miluju...) Ale kdybyste si i přes varování poranili nohu (nebo jiné části těla), pokuste se z toho poučit.

Poděkování

Díky za pomoc s tipy, opravami a tak:

[76] https://docs.python.org/2/library/re.html

- ldolse
- kovidgoyal
- chaley
- dwanthny
- kacir
- Starson17

Více o regulárních výrazech najdete v Uživatelské příručce Pythonu[77].

1.9.5 Integrace serveru s obsahem calibre do jiných serverů

Zde vám ukážeme, jak integrovat server s obsahem calibre do dalších serverů. Nejběžnějším důvodem k tomu je použít SSl nebo propracovanějšího ověřování. Jsou dva hlavní postupy: Spuštění serveru s obsahem calibre jako samostatného procesu a použití reverzní proxy k propojení s vaším hlavním serverem, nebo spuštění serveru s obsahem v procesu ve vašem hlavním serveru s WSGI. Příklady níže jsou všechny pro Apache 2.x na Linuxu, ale měly by být snadno přizpůsobitelný na jiné platformy.

Obsah

- *Používání reverzní proxy* (stránka 168)
- *V rámci procesu* (stránka 169)

Poznámka: Toto platí pouze pro vydání calibre >= 0.7.25

Používání reverzní proxy

Reverzní proxy je, když váš normální server přijímá příchozí požadavky a předává je serveru Calibre. Pak přečte odpověď od serveru Calibre a předá ji klientovi. To znamená, že můžete jednoduše spustit server Calibre jako normálně, aniž byste se ho snažili úzce integrovat s vaším hlavním serverem, a můžete využít jakékoliv příhodné systémy ověřování, které má váš hlavní server. Toto je nejjednodušší přístup, protože vám umožňuje použít binární instalaci Calibre bez žádných požadavků na externí závislosti nebo systémovou integraci. Níže je příklad, jak toho dosáhnout s Apache jako vaším hlavním serverem, ale bude to fungovat s jakýmkoliv serverem, který podporuje reverzní proxy.

Nejdříve spusťte server s obsahem calibre, jak je zobrazeno níže:

```
calibre-server --url-prefix /calibre --port 8080
```

Klíčový parametr je zde `--url-prefix /calibre`. Toto způsobí, že server s obsahem bude sloužit všem URL s předponou calibre. Abyste to viděli v akci, navštivte ve svém prohlížeči `http://localhost:8080/calibre`. Měli byste vidět normální webovou stránku serveru s obsahem, ale nyní poběží pod /calibre.

Nyní předpokládejme, že používáte Apache jako svůj hlavní server. Nejdříve povolte moduly proxy v Apache přidáním následujícího do `httpd.conf`:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Přesný postup povolení modulů proxy se bude lišit v závislosti na vaší instalaci Apache. Jakmile budete mít povolené moduly proxy, přidejte následující pravidla do httpd.conf (nebo pokud používáte virtuální hostitele, tak do konfiguračního souboru příslušného virtuálního hostitele):

[77] https://docs.python.org/2/library/re.html

```
RewriteEngine on
RewriteRule ^/calibre/(.*) http://localhost:8080/calibre/$1 [proxy]
RewriteRule ^/calibre http://localhost:8080 [proxy]
SetEnv force-proxy-request-1.0 1
SetEnv proxy-nokeepalive 1
```

To je vše, nyní budete schopni přistupovat k serveru s obsahem calibre pod URL /calibre ve svém serveru Apache. Výše zmíněná pravidla předají všechny požadavky pod /calibre serveru calibre běžícím na portu 8080 a díky výše zmíněné volbě –url-prefix je server calibre zpracuje transparentně.

Poznámka: Pokud chcete věnovat celého virtuálního hostitele serveru s obsahem, není potřeba používat –url-prefix a RewriteRule, namísto toho použijte pouze direktivu ProxyPass.

Poznámka: Modul serveru používaný calibre, CherryPy, může mít potíže s používáním proxy a požadavky KeepAlive, takže je v Apache vypněte pomocí výše zobrazené direktivy SetEnv.

V rámci procesu

Server s obsahem calibre může být spuštěn přímo, v rámci procesu, uvnitř serveru hostitel, jako je Apache, pomocí rozhraní WSGI.

Poznámka: Aby toto fungovalo, musí být ve vašem systému nainstalovány všechny závislosti vyžadované calibre. To je velice netriviální a je doporučeno, abyste nepoužívali servery v rámci procesu. Nedostane se vám žádné pomoci s laděním problémů serveru v rámci procesu.

Nejdříve musíme vytvořit *adaptér* WSGI pro server s obsahem calibre. Zde je šablona, kterou můžete k tomuto účelu použít. Nahraďte cesty, jak je doporučeno v komentářích

```
# WSGI script file to run calibre content server as a WSGI app

import sys, os


# You can get the paths referenced here by running
# calibre-debug --paths
# on your server

# The first entry from CALIBRE_PYTHON_PATH
sys.path.insert(0, '/home/kovid/work/calibre/src')

# CALIBRE_RESOURCES_PATH
sys.resources_location = '/home/kovid/work/calibre/resources'

# CALIBRE_EXTENSIONS_PATH
sys.extensions_location = '/home/kovid/work/calibre/src/calibre/plugins'

# Path to directory containing calibre executables
sys.executables_location = '/usr/bin'

# Path to a directory for which the server has read/write permissions
# calibre config will be stored here
os.environ['CALIBRE_CONFIG_DIRECTORY'] = '/var/www/localhost/calibre-config'

del sys
```

```
del os

from calibre.library.server.main import create_wsgi_app
application = create_wsgi_app(
        # The mount point of this WSGI application (i.e. the first argument to
        # the WSGIScriptAlias directive). Set to empty string is mounted at /
        prefix='/calibre',

        # Path to the calibre library to be served
        # The server process must have write permission for all files/dirs
        # in this directory or BAD things will happen
        path_to_library='/home/kovid/documents/demo library',

        # The virtual library (restriction) to be used when serving this
        # library.
        virtual_library=None
)

del create_wsgi_app
```

Uložte tento adaptér jako `calibre-wsgi-adpater.py` někam, kde k němu bude mít váš server přístup.

Předpokládejme, že chceme použít WSGI v Apache. Nejdříve povolte WSGI v Apache přidáním následujícího do `httpd.conf`:

```
LoadModule wsgi_module modules/mod_wsgi.so
```

Přesný postup povolení modulu wsgi se bude lišit v závislosti na vaší instalaci Apache. Jakmile budete mít povolené moduly proxy, přidejte následující pravidla do httpd.conf (nebo pokud používáte virtuální hostitele, tak do konfiguračního souboru příslušného virtuálního hostitele):

```
WSGIScriptAlias /calibre /var/www/localhost/cgi-bin/calibre-wsgi-adapter.py
```

Změňte cestu k `calibre-wsgi-adapter.py` podle toho, kam jste ho dříve uložili (ujistěte se, že k němu má Apache přístup).

To je vše, nyní budete schopni přistupovat k serveru s obsahem calibre pod URL /calibre ve svém serveru Apache.

Poznámka: Pro další nápovědu k používání mod_wsgi v Apache se podívejte na mod_wsgi[78].

1.9.6 Psaní vlastních modulů pro rozšíření funkcí Calibre

Calibre má velice modulární provedení. Téměř všechny funkce jsou ve formě modulů. Moduly jsou použity pro převod, stahování zpráv (ačkoliv ty jsou nazvány předpisy), různé komponenty uživatelského rozhraní, připojení k různým zařízením, zpracování souborů při jejich přidávání do calibre a tak dále. Kompletní seznam všech vestavěných modulů můžete získat v |calibre| v *Předvolby->Moduly*.

Zde vás naučíme, jak vytvořit vlastní moduly pro přidání nových funkcí do Calibre.

[78] http://code.google.com/p/modwsgi/wiki/WhereToGetHelp

Obsah

- *Anatomie modulu Calibre* (stránka 171)
- *Modul uživatelského rozhraní* (stránka 172)
  - *__init__.py* (stránka 173)
  - *ui.py* (stránka 174)
  - *main.py* (stránka 175)
  - *Získání zdrojů ze souboru zip modulu* (stránka 178)
  - *Povolení uživatelské konfigurace vašeho modulu* (stránka 178)
- *Moduly pro úpravu knihy* (stránka 180)
  - *main.py* (stránka 180)
- *Přidávání překladů do vašeho modulu* (stránka 183)
- *Rozhraní API modulu* (stránka 183)
- *Ladění modulů* (stránka 183)
- *Další příklady modulů* (stránka 184)
- *Sdílení vašeho modulu s ostatními* (stránka 184)

Poznámka: Toto platí pouze pro vydání Calibre >= 0.8.60

Anatomie modulu Calibre

Modul Calibre je velice jednoduchý, je to prostě soubor zip, který obsahuje nějaký kód Pythonu a jakékoliv jiné zdroje, jako soubory obrázků vyžadované modulem. Bez dalších okolků se pojďme podívat na základní příklad.

Předpokládejme, že máte instalaci Calibre, kterou používáte pro vlastní publikaci různých elektronických dokumentů ve formátech EPUB a MOBI. Chtěli byste, aby všechny soubory generované v Calibre měli vydavatele nastaveného jako "Hello World", zde je návod, jak to udělat. Vytvořte soubor s názvem `__init__.py` (toto je speciální název a musí být vždy použit pro hlavní soubor vašeho modulu) a zadejte do něj následující kód Pythonu:

```
import os
from calibre.customize import FileTypePlugin

class HelloWorld(FileTypePlugin):

    name                = 'Hello World Plugin' # Name of the plugin
    description         = 'Set the publisher to Hello World for all new conversions'
    supported_platforms = ['windows', 'osx', 'linux'] # Platforms this plugin will run on
    author              = 'Acme Inc.' # The author of this plugin
    version             = (1, 0, 0)   # The version number of this plugin
    file_types          = set(['epub', 'mobi']) # The file types that this plugin will be applied to
    on_postprocess      = True # Run this plugin after conversion is complete
    minimum_calibre_version = (0, 7, 53)

    def run(self, path_to_ebook):
        from calibre.ebooks.metadata.meta import get_metadata, set_metadata
        file = open(path_to_ebook, 'r+b')
        ext  = os.path.splitext(path_to_ebook)[-1][1:].lower()
        mi = get_metadata(file, ext)
        mi.publisher = 'Hello World'
        set_metadata(file, mi, ext)
        return path_to_ebook
```

To je vše. Pro přidání tohoto kódu do Calibre jako modulu stačí spustit následující v adresáři, ve kterém jste vytvořili

__init__.py:

```
calibre-customize -b .
```

Poznámka: Na OS X jsou nástroje příkazového řádku uvnitř sady calibre, například pokud jste nainstalovali Calibre do /Applications, nástroje příkazového řádku jsou v /Applications/calibre.app/Contents/console.app/Contents/MacOS

Modul Hello World můžete stáhnout z helloworld_plugin.zip[79].

Pokaždé, když použijete Calibre, aby převedlo knihu, bude zavolána metoda run() modulu a převedená kniha bude mít vydavatele nastaveného na "Hello World". Toto je jednoduchý modul, pojďme se přesunout na složitější příklad, který skutečně přidá komponentu do uživatelského rozhraní.

Modul uživatelského rozhraní

Tento modul bude rozdělen do několika souborů (aby kód zůstal čistý). Ukáže vám, jak získat zdroje (soubory obrázků nebo dat) ze souboru zip modulu, což umožňuje uživatelům konfigurovat váš modul, jak vytvořit prvky v uživatelském rozhraní Calibre a jak přistupovat k a dotazovat se databáze knih v Calibre.

Tento modul můžete stáhnout z interface_demo_plugin.zip[80]. První věc, na kterou pamatujte, je to, že tento soubor zip má v sobě mnohem více souborů, vysvětleno níže, věnujte zvláštní pozornost plugin-import-name-interface_demo.txt.

plugin-import-name-interface_demo.txt Prázdný textový soubor použitý pro umožnění kouzla vícesouborového modulu. Tento soubor musí být přítomen ve všech modulech, které používají více než jeden soubor .py. Měl by být prázdný a jeho název souboru musí být ve formě: plugin-import-name-**some_name**.txt. Přítomnost tohoto souboru vám umožňuje importovat kód ze souborů .py přítomných uvnitř souboru zip pomocí příkazu, jako je:

```
from calibre_plugins.some_name.some_module import some_object
```

Předpona calibre_plugins musí být vždy přítomná. some_name pochází z názvu prázdného textového souboru. some_module odkazuje na soubor some_module.py uvnitř souboru zip. Pamatujte, že toto importování je právě tak výkonné jako obyčejné importy Pythonu. Uvnitř souboru zip můžete vytvářet balíčky a podbalíčky modulů .py, jako byste to dělali normálně (definováním __init__.py v každém podadresáři) a všechno by mělo prostě fungovat.

Název, který použijete pro some_name, zadá globální obor názvů sdílený všemi moduly, **takže by měl být co nejjedinečnější**. Ale nezapomeňte, že to musí být platný identifikátor Pythonu (pouze abecední znaky, číslice a podtržítko).

__init__.py Jako předtím, soubor, který definuje třídu modulu

main.py Tento soubor obsahuje skutečný kód, který dělá něco užitečného

ui.py Tento soubor definuje součást rozhraní modulu

images/icon.png Ikona pro tento modul

about.txt Textový soubor s informacemi o modulu

translations Složka obsahující soubory .mo s překlady uživatelského rozhraní vašeho modulu do různých jazyků. Podívejte se níže na podrobnosti.

Nyní se podívejme na kód.

[79] http://calibre-ebook.com/downloads/helloworld_plugin.zip
[80] http://calibre-ebook.com/downloads/interface_demo_plugin.zip

__init__.py

Nejdříve povinný `__init__.py` pro definování metadat modulu:

```
# The class that all Interface Action plugin wrappers must inherit from
from calibre.customize import InterfaceActionBase

class InterfacePluginDemo(InterfaceActionBase):
    '''
    This class is a simple wrapper that provides information about the actual
    plugin class. The actual interface plugin class is called InterfacePlugin
    and is defined in the ui.py file, as specified in the actual_plugin field
    below.

    The reason for having two classes is that it allows the command line
    calibre utilities to run without needing to load the GUI libraries.
    '''
    name                = 'Interface Plugin Demo'
    description         = 'An advanced plugin demo'
    supported_platforms = ['windows', 'osx', 'linux']
    author              = 'Kovid Goyal'
    version             = (1, 0, 0)
    minimum_calibre_version = (0, 7, 53)

    #: This field defines the GUI plugin class that contains all the code
    #: that actually does something. Its format is module_path:class_name
    #: The specified class must be defined in the specified module.
    actual_plugin       = 'calibre_plugins.interface_demo.ui:InterfacePlugin'

    def is_customizable(self):
        '''
        This method must return True to enable customization via
        Preferences->Plugins
        '''
        return True

    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # It is important to put this import statement here rather than at the
        # top of the module as importing the config class will also cause the
        # GUI libraries to be loaded, which we do not want when using calibre
        # from the command line
```

```
        from calibre_plugins.interface_demo.config import ConfigWidget
        return ConfigWidget()

    def save_settings(self, config_widget):
        '''
        Save the settings specified by the user with config_widget.

        :param config_widget: The widget returned by :meth:`config_widget`.
        '''
        config_widget.save_settings()

        # Apply the changes
        ac = self.actual_plugin_
        if ac is not None:
            ac.apply_settings()
```

Jediná funkce, která stojí za pozornost, je pole `actual_plugin`. Protože Calibre má jak příkazový řádek, tak grafická rozhraní, moduly grafického rozhraní, jako je tento, by v __init__.py neměly načítat žádné knihovny grafického rozhraní. Pole actual_plugin to dělá za vás tím, že řekne Calibre, že skutečný modul je třeba hledat v jiném souboru uvnitř vašeho archivu zip, který bude načten pouze v kontextu grafického rozhraní.

Nezapomeňte, že aby toto fungovalo, musíte mít ve svém souboru zip soubor plugin-import-name-some_name.txt, jak bylo probráno výše.

Také existuje pár metod pro povolení uživatelské konfigurace modulu. Ty jsou popsány níže.

ui.py

Nyní se podívejme na ui.py, který definuje skutečné grafické rozhraní modulu. Zdrojový kód je silně okomentovaný a měl by být sebevysvětlující:

```
# The class that all interface action plugins must inherit from
from calibre.gui2.actions import InterfaceAction
from calibre_plugins.interface_demo.main import DemoDialog

class InterfacePlugin(InterfaceAction):

    name = 'Interface Plugin Demo'

    # Declare the main action associated with this plugin
    # The keyboard shortcut can be None if you dont want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,
            'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

    def genesis(self):
        # This method is called once per plugin, do initial setup here

        # Set the icon for this interface action
        # The get_icons function is a builtin function defined for all your
        # plugin code. It loads icons from the plugin zip file. It returns
        # QIcon objects, if you want the actual data, use the analogous
        # get_resources builtin function.
        #
        # Note that if you are loading more than one icon, for performance, you
```

```
        # should pass a list of names to get_icons. In this case, get_icons
        # will return a dictionary mapping names to QIcons. Names that
        # are not found in the zip file will result in null QIcons.
        icon = get_icons('images/icon.png')

        # The qaction is automatically created from the action_spec defined
        # above
        self.qaction.setIcon(icon)
        self.qaction.triggered.connect(self.show_dialog)

    def show_dialog(self):
        # The base plugin object defined in __init__.py
        base_plugin_object = self.interface_action_base_plugin
        # Show the config dialog
        # The config dialog can also be shown from within
        # Preferences->Plugins, which is why the do_user_config
        # method is defined on the base plugin class
        do_user_config = base_plugin_object.do_user_config

        # self.gui is the main calibre GUI. It acts as the gateway to access
        # all the elements of the calibre user interface, it should also be the
        # parent of the dialog
        d = DemoDialog(self.gui, self.qaction.icon(), do_user_config)
        d.show()

    def apply_settings(self):
        from calibre_plugins.interface_demo.config import prefs
        # In an actual non trivial plugin, you would probably need to
        # do something based on the settings in prefs
        prefs
```

main.py

Skutečná logika pro implementaci dialogového okna rozhraní ukázkového modulu.

```
from PyQt5.Qt import QDialog, QVBoxLayout, QPushButton, QMessageBox, QLabel

from calibre_plugins.interface_demo.config import prefs

class DemoDialog(QDialog):

    def __init__(self, gui, icon, do_user_config):
        QDialog.__init__(self, gui)
        self.gui = gui
        self.do_user_config = do_user_config

        # The current database shown in the GUI
        # db is an instance of the class LibraryDatabase from db/legacy.py
        # This class has many, many methods that allow you to do a lot of
        # things. For most purposes you should use db.new_api, which has
        # a much nicer interface from db/cache.py
        self.db = gui.current_db

        self.l = QVBoxLayout()
        self.setLayout(self.l)
```

```
        self.label = QLabel(prefs['hello_world_msg'])
        self.l.addWidget(self.label)

        self.setWindowTitle('Interface Plugin Demo')
        self.setWindowIcon(icon)

        self.about_button = QPushButton('About', self)
        self.about_button.clicked.connect(self.about)
        self.l.addWidget(self.about_button)

        self.marked_button = QPushButton(
            'Show books with only one format in the calibre GUI', self)
        self.marked_button.clicked.connect(self.marked)
        self.l.addWidget(self.marked_button)

        self.view_button = QPushButton(
            'View the most recently added book', self)
        self.view_button.clicked.connect(self.view)
        self.l.addWidget(self.view_button)

        self.update_metadata_button = QPushButton(
            'Update metadata in a book\'s files', self)
        self.update_metadata_button.clicked.connect(self.update_metadata)
        self.l.addWidget(self.update_metadata_button)

        self.conf_button = QPushButton(
                'Configure this plugin', self)
        self.conf_button.clicked.connect(self.config)
        self.l.addWidget(self.conf_button)

        self.resize(self.sizeHint())

    def about(self):
        # Get the about text from a file inside the plugin zip file
        # The get_resources function is a builtin function defined for all your
        # plugin code. It loads files from the plugin zip file. It returns
        # the bytes from the specified file.
        #
        # Note that if you are loading more than one file, for performance, you
        # should pass a list of names to get_resources. In this case,
        # get_resources will return a dictionary mapping names to bytes. Names that
        # are not found in the zip file will not be in the returned dictionary.
        text = get_resources('about.txt')
        QMessageBox.about(self, 'About the Interface Plugin Demo',
                text.decode('utf-8'))

    def marked(self):
        ''' Show books with only one format '''
        db = self.db.new_api
        matched_ids = {book_id for book_id in db.all_book_ids() if len(db.formats(book_id)) == 1}
        # Mark the records with the matching ids
        # new_api does not know anything about marked books, so we use the full
        # db object
        self.db.set_marked_ids(matched_ids)

        # Tell the GUI to search for all marked records
        self.gui.search.setEditText('marked:true')
        self.gui.search.do_search()
```

```
    def view(self):
        ''' View the most recently added book '''
        most_recent = most_recent_id = None
        db = self.db.new_api
        for book_id, timestamp in db.all_field_for('timestamp', db.all_book_ids()).iteritems():
            if most_recent is None or timestamp > most_recent:
                most_recent = timestamp
                most_recent_id = book_id

        if most_recent_id is not None:
            # Get a reference to the View plugin
            view_plugin = self.gui.iactions['View']
            # Ask the view plugin to launch the viewer for row_number
            view_plugin._view_calibre_books([most_recent_id])

    def update_metadata(self):
        '''
        Set the metadata in the files in the selected book's record to
        match the current metadata in the database.
        '''
        from calibre.ebooks.metadata.meta import set_metadata
        from calibre.gui2 import error_dialog, info_dialog

        # Get currently selected books
        rows = self.gui.library_view.selectionModel().selectedRows()
        if not rows or len(rows) == 0:
            return error_dialog(self.gui, 'Cannot update metadata',
                                'No books selected', show=True)
        # Map the rows to book ids
        ids = list(map(self.gui.library_view.model().id, rows))
        db = self.db.new_api
        for book_id in ids:
            # Get the current metadata for this book from the db
            mi = db.get_metadata(book_id, get_cover=True, cover_as_data=True)
            fmts = db.formats(book_id)
            if not fmts:
                continue
            for fmt in fmts:
                fmt = fmt.lower()
                # Get a python file object for the format. This will be either
                # an in memory file or a temporary on disk file
                ffile = db.format(book_id, fmt, as_file=True)
                ffile.seek(0)
                # Set metadata in the format
                set_metadata(ffile, mi, fmt)
                ffile.seek(0)
                # Now replace the file in the calibre library with the updated
                # file. We dont use add_format_with_hooks as the hooks were
                # already run when the file was first added to calibre.
                db.add_format(book_id, fmt, ffile, run_hooks=False)

        info_dialog(self, 'Updated files',
                'Updated the metadata in the files of %d book(s)'%len(ids),
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
```

```
        self.label.setText(prefs['hello_world_msg'])
```

Získání zdrojů ze souboru zip modulu

Systém načítání doplňků Calibre definuje pár vestavěných funkcí, které vám umožňují pohodlně získat soubory ze souboru zip modulu.

get_resources(name_or_list_of_names) Tato funkce by měla být volána se seznamem cest k souborům uvnitř souboru zip. Například pro přístup k souboru icon.png v adresáři images v souboru zip byste použili: `images/icon.png`. Vždy používejte lomítko jako oddělovač cesty, dokonce i ve Windows. Při předání jednoho názvu vrátí funkce nezpracované bajty tohoto souboru nebo None, pokud název nebyl v souboru zip nalezen. Pokud předáte více než jeden název, pak vrací slovník mapující názvy na bajty. Pokud název není nalezen, nebude přítomen ve vráceném slovníku.

get_icons(name_or_list_of_names) Pohodlný obal pro get_resources(), který vytvoří objekty QIcon z nezpracovaných bajtů vrácených get_resources. Pokud název není v souboru zip nalezen, odpovídající QIcon bude nulový.

Povolení uživatelské konfigurace vašeho modulu

Abyste povolili uživatelům konfigurovat váš modul, musíte definovat ve své základní třídě modulu tři metody, '**is_customizable**, **config_widget** a **save_settings** jak je uvedeno níže:

```
    def is_customizable(self):
        '''
        This method must return True to enable customization via
        Preferences->Plugins
        '''
        return True
```

```
    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
        displayed as a warning dialog to the user and the process will be
        aborted.

        The base class implementation of this method raises NotImplementedError
        so by default no user configuration is possible.
        '''
        # It is important to put this import statement here rather than at the
        # top of the module as importing the config class will also cause the
        # GUI libraries to be loaded, which we do not want when using calibre
        # from the command line
        from calibre_plugins.interface_demo.config import ConfigWidget
        return ConfigWidget()
```

```
    def save_settings(self, config_widget):
        '''
        Save the settings specified by the user with config_widget.

        :param config_widget: The widget returned by :meth:`config_widget`.
        '''
        config_widget.save_settings()

        # Apply the changes
        ac = self.actual_plugin_
        if ac is not None:
            ac.apply_settings()
```

Calibre má mnoho různých způsobů, jak ukládat konfigurační data (starší verze jeho dlouhé historie). Doporučený způsob je použít třídu **JSONConfig**, která ukládá vaše konfigurační informace v souboru .json.

Kód pro správu konfiguračních dat v ukázkovém modulu je v config.py:

```
from PyQt5.Qt import QWidget, QHBoxLayout, QLabel, QLineEdit

from calibre.utils.config import JSONConfig

# This is where all preferences for this plugin will be stored
# Remember that this name (i.e. plugins/interface_demo) is also
# in a global namespace, so make it as unique as possible.
# You should always prefix your config file name with plugins/,
# so as to ensure you dont accidentally clobber a calibre config file
prefs = JSONConfig('plugins/interface_demo')

# Set defaults
prefs.defaults['hello_world_msg'] = 'Hello, World!'

class ConfigWidget(QWidget):

    def __init__(self):
        QWidget.__init__(self)
        self.l = QHBoxLayout()
        self.setLayout(self.l)

        self.label = QLabel('Hello world &message:')
        self.l.addWidget(self.label)

        self.msg = QLineEdit(self)
        self.msg.setText(prefs['hello_world_msg'])
        self.l.addWidget(self.msg)
        self.label.setBuddy(self.msg)

    def save_settings(self):
        prefs['hello_world_msg'] = unicode(self.msg.text())
```

Object `prefs` je nyní dostupný v celém kódu modulu jednoduchým:

```
from calibre_plugins.interface_demo.config import prefs
```

Můžete vidět objekt `prefs` použitý v main.py:

Moduly pro úpravu knihy

Nyní pojďme na chvíli přeřadit a podívejme se na vytvoření modulu pro přidání nástrojů do editoru knih Calibre. Modul je dostupný zde: editor_demo_plugin.zip[81].

Prvním krokem, stejně jako pro všechny moduly, je vytvořit název importu prázdného souboru txt, jak je popsáno *výše* (stránka 172). Pojmenujeme soubor `plugin-import-name-editor_plugin_demo.txt`.

Nyní vytvoříme povinný soubor `__init__.py`, který obsahuje metadata o modulu – jeho název, autora, verzi atd.

```
from calibre.customize import EditBookToolPlugin


class DemoPlugin(EditBookToolPlugin):

    name = 'Edit Book plugin demo'
    version = (1, 0, 0)
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'osx', 'linux']
    description = 'A demonstration of the plugin interface for the ebook editor'
    minimum_calibre_version = (1, 46, 0)
```

Jeden modul editoru může poskytnout více nástrojů, každý nástroj odpovídá jednomu tlačítku na panelu nástrojů a položce v nabídce :guilabel:Moduly‘ v editoru. Ty mohou mít podnabídky v případě, že nástroj má více souvisejících akcí.

Nástroje musí být všechny definovány v souboru `main.py` ve vašem modulu. Každý nástroj je třída, která dědí ze třídy *`calibre.gui2.tweak_book.plugin.Tool`* (stránka 279). Pojďme se podívat na `main.py` z ukázkového modulu, zdrojový kód je silně komentován a měl by být samovysvětlující. Pro více informací si přečtěte dokumenty API ke třídě *`calibre.gui2.tweak_book.plugin.Tool`* (stránka 279).

main.py

Zde uvidíme definici jednoho nástroje, který se bude násobit všechny velikosti písem v knize číslem zadaným uživatelem. Tento nástroj ukazuje různé důležité pojmy, které budete potřebovat při vývoji svých vlastních modulů, takže byste si měli pečlivě přečíst (silně komentovaný) zdrojový kód.

```
import re
from PyQt5.Qt import QAction, QInputDialog
from cssutils.css import CSSRule

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool

from calibre import force_unicode
from calibre.gui2 import error_dialog
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize

class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True
```

[81] http://calibre-ebook.com/downloads/editor_demo_plugin.zip

```
    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui)  # noqa
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=('Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book will be multiplie
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
                import traceback
                error_dialog(self.gui, _('Failed to magnify fonts'), _(
                    'Failed to magnify fonts, click "Show details" for more info'),
                    det_msg=traceback.format_exc(), show=True)
                # Revert to the saved restore point
                self.boss.revert_requested(self.boss.global_undo.previous_container)
            else:
                # Show the user what changes we have made, allowing her to
                # revert them if necessary
                self.boss.show_current_diff()
                # Update the editor UI to take into account all the changes we
                # have made
                self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container  # The book being edited as a container object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.iteritems():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are cssutils CSSStylesheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name)  # Tell the container that we have changed the stylesheet
```

```
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_print=True)
                        container.dirty(name)  # Tell the container that we have changed the styleshe
                for elem in container.parsed(name).xpath('//*[@style]'):
                    # Process inline style attributes
                    block = container.parse_css(elem.get('style'), is_declaration=True)
                    self.magnify_declaration(block, factor)
                    elem.set('style', force_unicode(block.getCssText(separator=' '), 'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.
        for rule in sheet.cssRules.rulesOfType(CSSRule.STYLE_RULE):
            self.magnify_declaration(rule.style, factor)

    def magnify_declaration(self, style, factor):
        # Magnify all fonts in the specified style declaration by the specified
        # factor
        val = style.getPropertyValue('font-size')
        if not val:
            return
        # see if the font-size contains a number
        num = re.search(r'[0-9.]+', val)
        if num is not None:
            num = num.group()
            val = val.replace(num, '%f' % (float(num) * factor))
            style.setProperty('font-size', val)
        # We should also be dealing with the font shorthand property and
        # font sizes specified as non numbers, but those are left as exercises
        # for the reader
```

Pojďme si rozebrat `main.py`. Vidíme, že definuje jeden nástroj s názvem *Zvětšit písma*. Tento nástroj požádá uživatele o číslo a vynásobí všechny velikosti písem v knize tímto číslem.

První důležitá věc je název nástroje, který musíte nastavit na nějaký relativně jedinečný řetězec, protože bude použit jako klíč pro tento nástroj.

Dalším důležitým vstupním bodem je *calibre.gui2.tweak_book.plugin.Tool.create_action()* (stránka 280). Tato metoda vytvoří objekty QAction, které se zobrazí v panelu nástrojů modulů a nabídce modulu. Také to volitelně přiřadí klávesovou zkratku, kterou si uživatel může přizpůsobit. Spuštěný signál z QAction je spojený s metodou ask_user(), která se zeptá uživatele na násobitel velikosti písma, a pak spustí kód zvětšení.

Kód zvětšení kód je dobře okomentován a poměrně jednoduchý. Hlavními věcmi, na které pamatujte, je to, že jste získali odkaz na okno editoru jako `self.gui` a *Boss* editoru jako `self.boss`. *Boss* je objekt, který řídí uživatelské rozhraní editoru. Má mnoho užitečných metod, které jsou popsány ve třídě *calibre.gui2.tweak_book.boss.Boss* (stránka 281).

Nakonec je zde `self.current_container`, což je odkaz na knihu upravovanou jako objekt *calibre.ebooks.oeb.polish.container.Container* (stránka 273). Znázorňuje knihu jako kolekci jejích základních souborů HTML, CSS nebo obrázků a má pohodlné metody pro provádění mnoha užitečných věcí. Objekt kontejneru a různé užitečné funkce nástrojů, které mohou být znovu použity v kódu vašeho modulu, jsou

popsány v *API Documentation for the ebook editing tools* (stránka 273).

Přidávání překladů do vašeho modulu

Všechny řetězce uživatelského rozhraní ve svém modulu můžete mít přeloženy a zobrazeny v jakémkoliv jazyce, který je nastaven pro hlavní uživatelské rozhraní Calibre.

Prvním krokem je projít zdrojový kód svého modulu a označit všechny řetězce viditelné uživatelem jako přeložitelné jejich obklopením _(). Například:

```
action_spec = (_('My plugin'), None, _('My plugin is cool'), None)
```

Pak použijte nějaký program pro vygenerování souborů .po ze zdrojového kódu vašeho doplňku. Měl by být jeden soubor .po pro každý jazyk, do kterého chcete překládat. Například: cs.po pro češtinu, de.po pro němčinu, fr.po pro francouzštinu a tak dále. K tomu můžete použít program Poedit[82].

Tyto soubory .po pošlete svým překladatelům. Jakmile je dostanete zpátky, zkompilovat je do souborů .mo. K tomu můžete znovu použít Poedit, nebo prostě udělejte:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

Soubory .mo umístěte do složky `translations` ve svém modulu.

Posledním krokem je jednoduše zavolat funkci *load_translations()* na začátku souborů .py vašeho modulu. Z výkonnostních důvodů byste měli volat tuto funkci pouze v těch souborech .py, které skutečně mají přeložitelné řetězce. Takže v typickém modulu uživatelského rozhraní byste ji volali na začátku `ui.py` ale ne `__init__.py`.

Překlady svých modulů můžete otestovat změnou jazyka uživatelského rozhraní v Calibre pod Předvolby->Vzhled nebo spuštěním Calibre takto:

```
CALIBRE_OVERRIDE_LANG=de calibre
```

Nahraďte `de` kódem jazyka, který chcete otestovat.

Rozhraní API modulu

Jak jste si možná všimli výše, modul Calibre je třída. V Calibre existují různé třídy pro různé typy modulů. Podrobnosti o každé třídě, včetně základní třídy všech modulů, lze nalézt v *API Documentation for plugins* (stránka 193).

Váš modul téměř určitě bude používat kód z Calibre. Abyste zjistili, jak najít různé kousky funkcí v základním kódu Calibre, přečtěte si část o *Rozvržení kódu* (stránka 260) Calibre.

Ladění modulů

Prvním a nejdůležitějším krokem je spuštění Calibre v režimu ladění. To můžete udělat z příkazového řádku pomocí:

```
calibre-debug -g
```

Nebo z Calibre kliknutím pravým tlačítkem myši na tlačítko Předvolby nebo pomocí klávesové zkratky *Ctrl+Shift+R*.

Při spuštění z příkazového řádku, bude výstup ladění vypsán do konzole, při spuštění z Calibre, půjde výstup do textového souboru.

Do kódu svého modulu můžete kamkoliv vložit příkazy tisku, budou mít výstup v režimu ladění. Nezapomeňte, že je to Python, neměli byste opravdu potřebovat nic víc než příkazy tisku pro ladění ;) Celé Celibre jsem vytvořil pouze pomocí této techniky ladění.

[82] http://poedit.net/

Změny ve svém modulu můžete rychle testovat pomocí následujícího příkazového řádku:

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/directory; calibre
```

Toto vypne spuštěné Calibre, počkejte až bude vypnutí dokončeno, a pak aktualizujte svůj modul v Calibre a znovu spusťte Calibre.

Další příklady modulů

Seznam mnoha propracovaných modulů Calibre můžete nalézt zde[83].

Sdílení vašeho modulu s ostatními

Pokud byste chtěli sdílet moduly, které jste vytvořili, s ostatními uživateli Calibre, zveřejněte svůj modul v novém vláknu ve fóru modulů Calibre[84].

1.9.7 Sazba matematiky v e-knihách

Prohlížeč e-knih calibre má schopnost zobrazit matematiku vloženou do e-knih (soubory ePub a HTML). Můžete sázet matematiku přímo pomocí TeX, MathML nebo AsciiMath. Prohlížeč calibre k tomuto používá vynikající knihovnu MathJax[85]. Toto je stručný kurz vytváření e-knih s matematikou, které fungují dobře s prohlížečem calibre.

Zjednodušený soubor HTML s matematikou

Můžete zapsat matematiku vloženou uvnitř souboru zjednodušeného HTML a prohlížeč calibre ji vykreslí se správnou sazbou matematiky. V příkladu níže používáme pro matematiku zápis TeX. Uvidíte, že můžete použít normální příkazy TeX, s malým upozorněním, že ampersandy a znaky menší než a větší než musí být zapsány jako & < a > v uvedeném pořadí.

První krok je říct calibre, že toto bude obsahovat matematiku. To uděláte přidáním následujícího fragmentu kódu do oddílu <head> souboru HTML:

```
<script type="text/x-mathjax-config"></script>
```

To je vše, nyní můžete psát matematiku, jak byste to udělali v souboru .tex. Například zde jsou Lorentzovy rovnice:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp;= \sigma(y-x) \\
\dot{y} &amp;= \rho x - y - xz \\
\dot{z} &amp;= -\beta z + xy
\end{align}
</p>
```

Tento fragment vypadá jako následující snímek obrazovky v prohlížeči calibre.

Kompletní soubor HTML s více rovnicemi a vloženou matematikou je reprodukován níže. Tento soubor HTML můžete převést do EPUB v calibre, abyste získali e-knihu, kterou můžete snadno distribuovat ostatním lidem.

[83] http://www.mobileread.com/forums/showthread.php?t=118764
[84] http://www.mobileread.com/forums/forumdisplay.php?f=237
[85] http://www.mathjax.org

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Obr. 1.2: *Lorentzovy rovnice*

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
<head>
<title>Math Test Page</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<!-- This script tag is needed to make calibre's ebook-viewer recpgnize that this file needs math typ
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly suppressed.
    MathJax.Hub.Config({ TeX: { equationNumbers: {autoNumber: "all"} } });
</script>

<style>
h1 {text-align:center}
h2 {
  font-weight: bold;
  background-color: #DDDDDD;
  padding: .2em .5em;
  margin-top: 1.5em;
  border-top: 3px solid #666666;
  border-bottom: 2px solid #999999;
}
</style>
</head>
<body>

<h1>Sample Equations</h1>

<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} &amp; = \sigma(y-x) \label{lorenz}\\
\dot{y} &amp; = \rho x - y - xz \\
\dot{z} &amp; = -\beta z + xy
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\!\!2} \leq
 \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)
\]</p>
```

```
<h2>A Cross Product Formula</h2>

<p>\[
  \mathbf{V}_1 \times \mathbf{V}_2 =
   \begin{vmatrix}
    \mathbf{i} &amp; \mathbf{j} &amp; \mathbf{k} \\
    \frac{\partial X}{\partial u} &amp; \frac{\partial Y}{\partial u} &amp; 0 \\
    \frac{\partial X}{\partial v} &amp; \frac{\partial Y}{\partial v} &amp; 0 \\
   \end{vmatrix}
\]</p>

<h2>The probability of getting \(k\) heads when flipping \(n\) coins is:</h2>

<p>\[P(E) = {n \choose k} p^k (1-p)^{ n-k} \]</p>

<h2>An Identity of Ramanujan</h2>

<p>\[
   \frac{1}{(\sqrt{\phi \sqrt{5}}-\phi) e^{\frac25 \pi}} =
     1+\frac{e^{-2\pi}} {1+\frac{e^{-4\pi}} {1+\frac{e^{-6\pi}}
      {1+\frac{e^{-8\pi}} {1+\ldots} } } }
\]</p>

<h2>A Rogers-Ramanujan Identity</h2>

<p>\[
  1 +  \frac{q^2}{(1-q)}+\frac{q^6}{(1-q)(1-q^2)}+\cdots =
    \prod_{j=0}^{\infty}\frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
     \quad\quad \text{for $|q|&lt;1$}.
\]</p>

<h2>Maxwell's Equations</h2>

<p>
\begin{align}
  \nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial\vec{\mathbf{E}}}{\partial t} &amp; = \
  \nabla \cdot \vec{\mathbf{E}} &amp; = 4 \pi \rho \\
  \nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial\vec{\mathbf{B}}}{\partial t} &amp; =
  \nabla \cdot \vec{\mathbf{B}} &amp; = 0
\end{align}
</p>

<h2>In-line Mathematics</h2>

<p>While display equations look good for a page of samples, the
ability to mix math and text in a paragraph is also important.  This
expression \(\sqrt{3x-1}+(1+x)^2\) is an example of an inline equation.  As
you see, equations can be used this way as well, without unduly
disturbing the spacing between lines.</p>

<h2>References to equations</h2>

<p>Here is a reference to the Lorenz Equations (\ref{lorenz}). Clicking on the equation number will t

</body>
</html>
```

Více informací

Protože prohlížeč calibre používá pro vykreslení matematiky knihovnu MathJax, nejlepší místo pro zjištění více o matematice v e-knihách a získání nápovědy je Webová stránka MathJax[86].

1.9.8 Vytváření katalogů AZW3 • EPUB • MOBI

Funkce calibre Vytvořit katalog vám umožňuje vytvořit z vaší knihovny katalog v různých formátech. Tento soubor nápovědy popisuje volby zařazení do katalogu při generování katalogu ve formátech AZW3, EPUB a MOBI.

- *Výběr knih do katalogu* (stránka 187)
- *Zahrnuté oddíly* (stránka 188)
- *Předpony* (stránka 188)
- *Vyloučené knihy* (stránka 189)
- *Vyloučené žánry* (stránka 189)
- *Další volby* (stránka 190)
- *Vlastní obálky katalogu* (stránka 191)
- *Dodatečné zdroje nápovědy* (stránka 191)

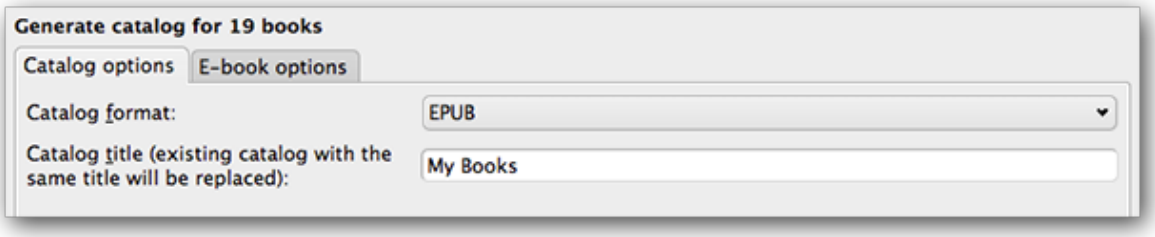
Výběr knih do katalogu

Pokud chcete katalogizovat *celou* svou knihovnu, odeberte v hlavním okně jakákoliv kritéria hledání nebo filtrování. Pokud máte vybránu jednu knihu, všechny knihy ve vaší knihovně budou kandidáti na zahrnutí do generovaného katalogu. Jednotlivé knihy mohou být vyloučeny různými kritérii, pro více informací se podívejte na oddíl *Vyloučené žánry* (stránka 189) níže.

Pokud chcete katalogizovat pouze *něco* z vaší knihovny, máte dvě volby:

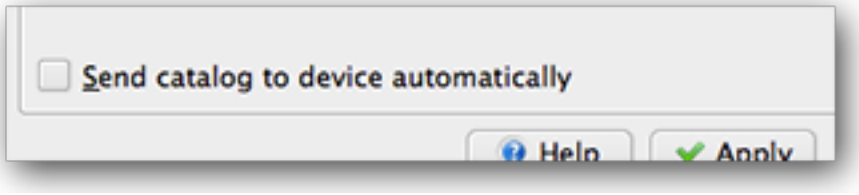
- Vytvořte výběr více knih, které chcete katalogizovat. Pokud je v hlavním okně calibre vybrána více než jedna kniha, budou katalogizovány pouze vybrané knihy.
- Použijte pole Hledat nebo Prohlížeč štítků pro filtrování zobrazených knih. Budou katalogizovány pouze vybrané knihy.

Pro zahájení generování katalogu vyberte položku nabídky *Převést knihy* > *Vytvořit katalog knih ve vaší knihovně calibre*.



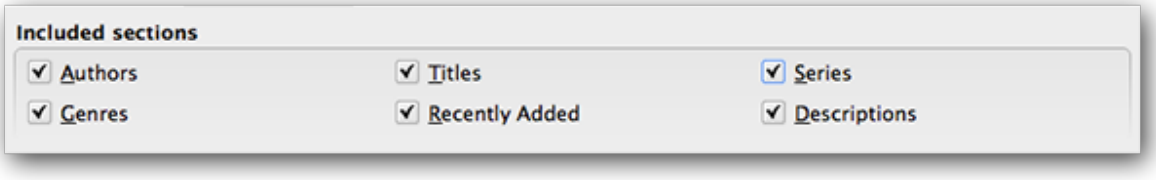
Ve *Volby katalogu* vyberte jako formát katalogu **AZW3, EPUB nebo MOBI**. V poli *Název katalogu* zadejte název, který bude použit pro vygenerovaný katalog. Pokud již existuje katalog se stejným názvem a formátem, bude nahrazen nově vytvořeným katalogem.

[86] http://www.mathjax.org



Povolení *Odeslat katalog do zařízení automaticky* stáhne po dokončení vygenerovaný katalog do připojeného zařízení.

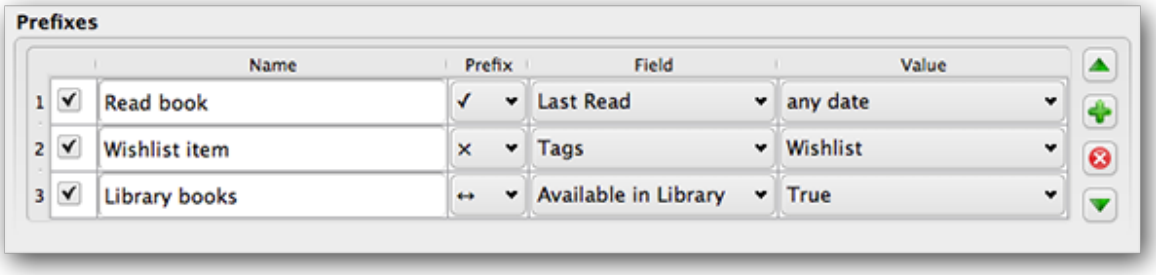
Zahrnuté oddíly



Oddíly povolené zatržítkem budou zahrnuty do generovaného katalogu:

- *Autoři* – všechny knihy seřazené podle autora zobrazené ve formátu seznamu. Knihy bez sérií jsou vypsány před knihami v sérii.
- *Názvy* – všechny knihy seřazené podle názvu zobrazené ve formátu seznamu.
- *Série* – všechny knihy, které jsou součástí série, seřazené podle názvu zobrazené ve formátu seznamu.
- *Žánry* – jednotlivé žánry zobrazené v seznamu, seřazené podle autora a série.
- *Nedávno přidané* – všechny knihy seřazené v obráceném chronologickém pořadí. Seznam obsahuje knihy přidané za posledních 30 dní, pak výpis přidaných knih po měsících.
- *Popisy* – podrobná stránka popisu pro každou knihu včetně náhledu obálky a komentářů. Seřazeno podle autora s knihami bez sérií vypsanými před knihami v sérii.

Předpony



Pravidla předpony vám umožňují přidat předponu do výpisů knih, když jsou splněna určitá kritéria. Například možná chcete označit přečtené knihy zatržítkem, nebo knihy na seznamu přání označit X.

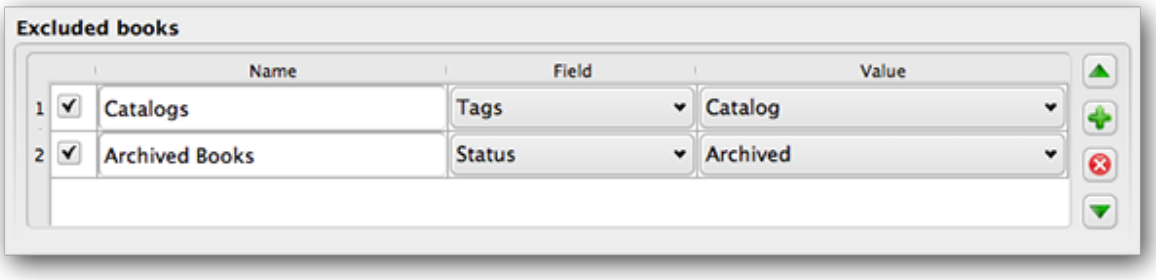
Zatržítko v prvním sloupci povolí pravidlo. *Název* je název pravidla, který zadáte. *Pole* je buď *Štítky*, nebo vlastní sloupec z vaší knihovny. *Hodnota* je obsah *Pole*, který porovnat. Když je pravidlo předpony splněno, kniha bude označena vybranou *Předponou*.

Vy výše uvedeném příkladu byla zadána tři pravidla předpony:

1. *Přečtená kniha* určuje, že kniha s jakýmkoliv datem ve vlastním sloupci pojmenovaném *Přečteno* bude označena předponou se symbolem zatržítka.
2. *Položka seznamu přání* určuje, že jakákoliv kniha se štítkem *Seznam přání* bude označena předponou se symbolem X.
3. *Knihy v knihovně* určuje, že jakákoliv kniha s hodnotou Pravda (nebo Ano) ve vlastním sloupci *Dostupná v knihovně* bude označena předponou se symbolem obousměrné šipky.

První odpovídající pravidlo určí předponu. Zakázaná nebo neúplná pravidla jsou ignorována.

Vyloučené knihy



Pravidla vyloučení vám umožňují zadat knihy, které nebudou katalogizovány.

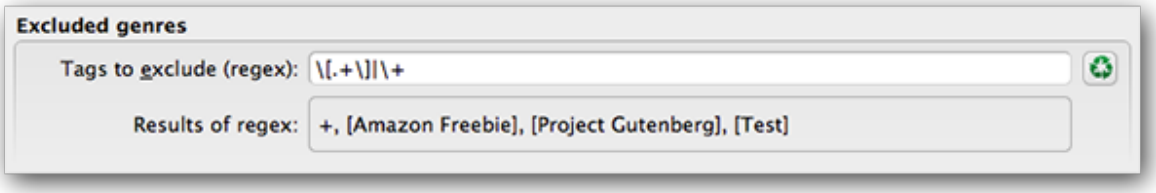
Zatržítko v prvním sloupci povolí pravidlo. *Název* je název pravidla, který zadáte. *Pole* je buď *Štítky*, nebo vlastní sloupec z vaší knihovny. *Hodnota* je obsah *Pole*, který porovnat. Když je pravidlo předpony splněno, kniha bude vyloučena z generovaného katalogu.

Vy výše uvedeném příkladu byla zadána dvě pravidla vyloučení:

1. Pravidlo *Katalogy* určuje, že jakákoliv kniha se štítkem *Katalog* bude vyloučena z generovaného katalogu.
2. Pravidlo *Archivované knihy* určuje, že jakákoliv kniha s hodnotou *Archivováno* ve vlastním sloupci guilabel:*Stav* bude vyloučena z generovaného katalogu.

Jsou vyhodnocována všechna pravidla pro každou knihu. Zakázaná nebo neúplná pravidla jsou ignorována.

Vyloučené žánry



Když je generován katalog, jsou štítky ve vaší databázi použity jako žánry. Například můžete použít štítky `Román` a `Literatura faktu`. Tyto štítky se stanou žánry ve vygenerovaném katalogu s knihami vypsanými pod jejich příslušnými seznamy žánrů na základě svých přiřazených štítků.

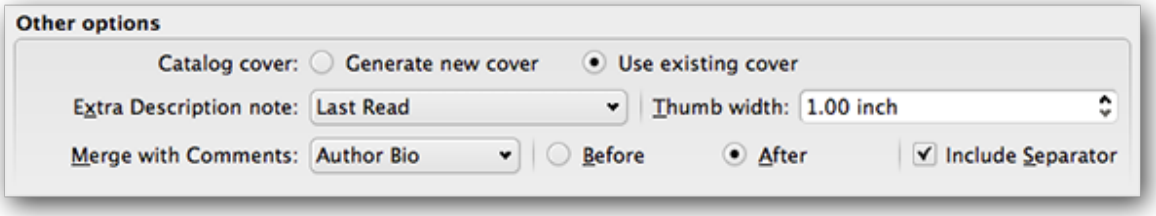
Možná používáte určité štítky pro jiné účely, možná + označení přečtené knihy, nebo štítek v hranatých závorkách, jako je `[Amazon Freebie]`, pro označení zdroje knihy. Regulární výraz *Vyloučené žánry* vám umožňuje zadat štítky,

které nechcete použít jako žánry v generovaném katalogu. Výchozí regulární výraz vyloučení `\[.+\]\+` vyloučí jakékoliv štítky ve formě `[štítek]`, stejně jako vyloučí `+`, výchozí štítek pro přečtené knihy, aby nebyly použity jako žánry v generovaném katalogu.

V regulárním výrazu můžete také použít přesný název štítku. Například `[Amazon Freebie]` nebo `[Projekt Gutenberg]`. Pokud chcete vypsat více přesných štítků k vyloučení, vložte mezi ně znak svislé čáry: `[Amazon Freebie]|[Projekt Gutenberg]`.

Výsledky regulárního výrazu zobrazí, které štítky budou vyloučeny při sestavování katalogu na základě štítků ve vaší databázi a regulárního výrazu, který jste zadali. Výsledky jsou aktualizovány, jak měníte regulární výraz.

Další volby



Obálka katalogu určuje, zda generovat novou obálku nebo použít existující obálku. Je možné vytvořit vlastní obálku pro vaše katalogy – pro více informací se podívejte na *Vlastní obálky katalogu* (stránka 191). Pokud jste vytvořili vlastní obálku, kterou chcete znovu použít, vyberte *Použít existující obálku*. Jinak vyberte *Generovat novou obálku*.

Dodatečná poznámka popisu určuje obsah vlastního sloupce, který vložit do stránky Popis vedle náhledu obálky. Například možná chcete zobrazit datum, kdy jste naposledy četli knihu, pomocí vlastního slupce *Naposledy čteno*. Pro pokročilé používání funkce poznámky Popisu se podívejte na tento příspěvek ve fóru calibre <http://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>‘_.

Šířka náhledu určuje předvolbu šířky pro náhledy obálek obsažené na stránkách Popis. Náhledy jsou ukládány do mezipaměti pro zlepšení výkonu. Pro experimentování s různými šířkami zkuste generovat katalog jen s pár knihami, doku neurčíte svou upřednostňovanou šířku, pak vygenerujte úplný katalog. Když je katalog generován poprvé s novou šířkou náhledu, výkon bude nižší, ale následující sestavení katalogu využijí náhledy v mezipaměti.

Sloučit s komentáři určuje vlastní sloupec, jehož obsah bude nedestruktivně sloučen s metadaty komentářů během generování katalogu. Například můžete mít vlastní sloupec *Biografie autora*, který byste chtěli připojit k metadatům komentářů. Můžete zvolit vložit obsah vlastního sloupce *před nebo za* oddíl Komentáře a volitelně oddělit připojený obsah vodorovnou čárou. Vhodné typy vlastních sloupců zahrnují `text, komentáře a složený`.

Vlastní obálky katalogu

S nainstalovaným modulem Generovat obálku[87] můžete vytvořit vlastní obálky pro své katalogy. Pro nainstalování modulu přejděte na *Předvolby* > *Pokročilé* > *Moduly* > *Získat nové moduly*.

Dodatečné zdroje nápovědy

Pro více informací o funkci Katalog calibre navštivte fórum MobileRead Creating Catalogs: Start here[88], kde můžete najít informace o tom, jak přizpůsobit šablony katalogu, a jak poslat chybové hlášení.

Pokud máte otázky nebo chcete diskutovat o funkci Katalog Calibre s ostatními uživateli, navštivte fórum MobileRead Calibre Catalogs[89].

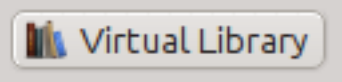
1.9.9 Virtuální knihovny

V calibre je virtuální knihovna způsobem, jak říct calibre, aby otevřelo pouze podmnožinu normální knihovny. Například možná chcete pracovat pouze s knihami od určitého autora, nebo s knihami majícími pouze určitý štítek. Používání virtuálních knihoven je upřednostňovaný způsob rozdělení vaší velké kolekce knih na menší podkolekce. Je to nadřazeno rozdělení vaší knihovny na více malých knihoven, protože když chcete prohledat celou svou kolekci, můžete jednoduše přejít zpět na plnou knihovnu. V není žádný způsob, jak souběžně prohledat více samostatných knihoven.

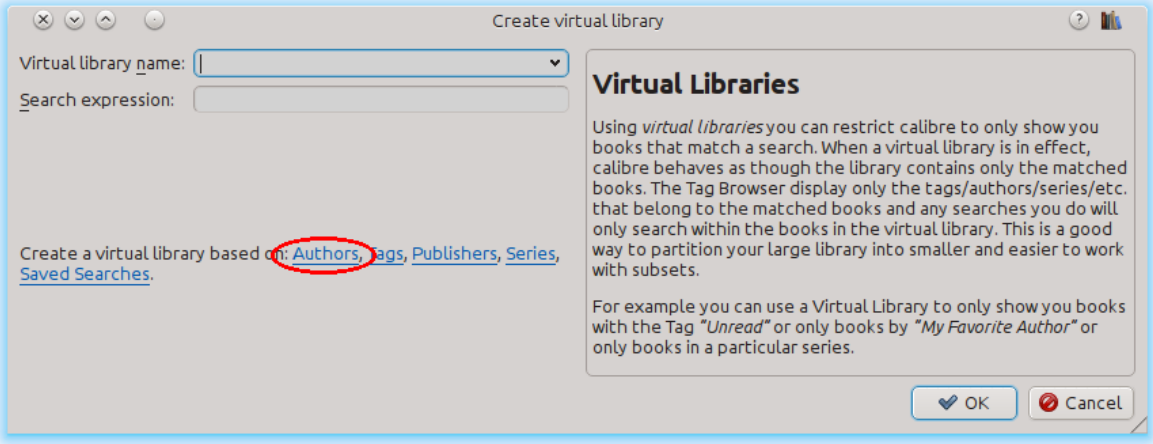
Virtuální knihovna se liší od jednoduchého hledání. Hledání pouze omezí seznam knih zobrazených v seznamu knih. Virtuální knihovna dělá toto a navíc také omezuje položky zobrazené v *Prohlížeči štítků* vlevo. Prohlížeč štítků zobrazí pouze štítky, autory, série, vydavatele atd., které pocházejí z knih ve virtuální knihovně. Virtuální knihovna se tudíž chová jako skutečná knihovna, ačkoliv obsahuje pouze omezenou sadu knih.

[87] http://www.mobileread.com/forums/showthread.php?t=124219
[88] http://www.mobileread.com/forums/showthread.php?t=118556
[89] http://www.mobileread.com/forums/forumdisplay.php?f=236

Vytváření virtuálních knihoven



Pro použití virtuální knihovny klikněte na tlačítko *Virtuální knihovna* umístěné vlevo od panelu hledání a vyberte volbu *Vytvořit virtuální knihovnu*. Jako první příklad vytvořme virtuální knihovnu, která zobrazuje pouze knihy od určitého autora. Klikněte na odkaz *Autoři*, jak je ukázáno na obrázku níže a zvolte autora, kterého chcete použít, a klikněte na OK.


Create virtual library
Virtual library name:
Search expression:
Create a virtual library based on: Authors, Tags, Publishers, Series, Saved Searches.
Virtual Libraries
Using virtual libraries you can restrict calibre to only show you books that match a search. When a virtual library is in effect, calibre behaves as though the library contains only the matched books. The Tag Browser display only the tags/authors/series/etc. that belong to the matched books and any searches you do will only search within the books in the virtual library. This is a good way to partition your large library into smaller and easier to work with subsets.
For example you can use a Virtual Library to only show you books with the Tag "Unread" or only books by "My Favorite Author" or only books in a particular series.
OK
Cancel

Dialogové okno Vytvořit virtuální knihovnu bylo pro vás vyplněno. Klikněte na OK a uvidíte, že byla vytvořena nová Virtuální knihovna a automaticky na ni bylo přepnuto, takže ukazuje pouze knihy od vybraného autora. Pokud jde o calibre, je to, jako kdyby vaše knihovna obsahovala pouze knihy od vybraného autora.

Na plnou knihovnu můžete kdykoliv přepnout zpět opětovným kliknutím na tlačítko *Virtuální knihovna* a výběrem položky pojmenované <*Žádná*>.

Virtuální knihovny jsou založeny na *hledání*. Jako základ virtuální knihovny můžete použít jakékoliv hledání. Virtuální knihovna bude obsahovat pouze knihy odpovídající tomuto hledání. Nejdříve vepište hledání, které chcete použít, do panelu hledání, nebo sestavte hledání pomocí *Prohlížeče štítků*. Až budete spokojení s vrácenými výsledky, klikněte na tlačítko Virtuální knihovna, zvolte Vytvořit virtuální knihovnu a zadejte název pro novou virtuální knihovnu. Virtuální knihovna pak bude vytvořena na základě hledání, které jste právě zadali. Hledání jsou velice výkonná, pro příklady druhů věcí, které s nimi můžete dělat, se podívejte na *Rozhraní hledání* (stránka 11).

Příklady užitečných Virtuálních knihoven

- **Knihy přidané do calibre za poslední den:** date:>1daysago
- **Knihy přidané do calibre za poslední měsíc:** date:>30daysago
- **Knihy s hodnocením 5 hvězdiček:** rating:5
- **Knihy s hodnocením alespoň 4 hvězdičky:** rating:>=4
- **Knihy bez hodnocení:** rating:false
- **Periodika stahovaná funkcí Načíst zprávy v calibre:** tags:=Zprávy and author:=calibre
- **Knihy bez štítků:** tags:false
- **Knihy bez obálek:** cover:false

Práce s Virtuální knihovnou

Upravit dříve vytvořenou virtuální knihovnu nebo ji odebrat můžete kliknutím na tlačítko *Virtuální knihovna* a zvolením příslušné akce.

Přejitím na *Předvolby->Chování* můžete říct calibre, že chcete vždy použít určitou virtuální knihovnu, když je otevřena aktuální knihovna.

Když používáte server s obsahem calibre, můžete ho nechat sdílet virtuální knihovnu namísto plné knihovny přejitím na *Předvolby->Sdílení po síti*.

Aktuální hledání můžete rychle použít jako dočasnou virtuální knihovnu kliknutím na tlačítko *Virtuální knihovna* a zvolením položky **aktuální hledání*.

Všechny virtuální knihovny můžete zobrazit jako karty nad seznamem knih. To je obzvlášť šikovné, pokud chcete velice často přepínat mezi virtuálními knihovnami. Klikněte na tlačítko *Virtuální knihovna* a vyberte *Zobrazit virtuální knihovny jako karty*. Karty můžete přeuspořádat přetažením a zavřením těch, které nechcete vidět. Zavřené karty mohou být obnoveny kliknutím pravým tlačítkem myši na panel karet.

Používání dodatečných omezení

Knihy zobrazené ve virtuální knihovně můžete dále omezit pomocí *Dodatečných omezení*. Dodatečné omezení je vámi dříve vytvořené uložené hledání, které může být použito na aktuální virtuální knihovnu pro další omezení zobrazených knih ve virtuální knihovně. Například řekněme, že máte virtuální knihovnu pro knihy se štítkem *Historický román* a uložené hledání, které vám ukazuje nepřečtené knihy. Můžete kliknout na tlačítko *Virtuální knihovna* a zvolit volbu *Dodatečná omezení* pro zobrazení pouze nepřečtených historických románů. Abyste se naučili o uložených hledáních, podívejte se na *Ukládání hledání* (stránka 13).

1.10 Přizpůsobení Calibre

Calibre má vysoce modulární provedení. Různé jeho části mohou být upraveny. Jak vytvořit *předpisy* pro přidání nových zdrojů online obsahu do Calibre se můžete naučit v oddílu *Přidávání oblíbených webových stránek se zprávami* (stránka 23). Zde se nejdříve naučíte, jak používat proměnné prostředí a *vylepšení* pro přizpůsobení chování Calibre, a potom jak zadat své vlastní statické zdroje, jako jsou ikony a šablony, pro přepsání výchozího nastavení a nakonec jak používat *moduly* pro přidávání funkcí do Calibre.

- *Proměnné prostředí* (stránka 217)
- *Vylepšení* (stránka 218)
- *Přepsání ikon, šablon atd.* (stránka 227)
- *Vytvoření vlastního motivu ikon pro Calibre* (stránka 228)
- *Přizpůsobení Calibre pomocí modulů* (stránka 228)

1.10.1 API Documentation for plugins

Defines various abstract base classes that can be subclassed to create powerful plugins. The useful classes are:

- *Plugin* (stránka 194)
- *FileTypePlugin* (stránka 196)
- *Metadata plugins* (stránka 196)
- *Catalog plugins* (stránka 197)
- *Metadata download plugins* (stránka 198)
- *Conversion plugins* (stránka 200)
- *Device Drivers* (stránka 202)
- *User Interface Actions* (stránka 212)
- *Preferences Plugins* (stránka 214)
- *Viewer plugins* (stránka 216)

Plugin

class calibre.customize.**Plugin**(*plugin_path*)

A calibre plugin. Useful members include:

- **self.plugin_path: Stores path to the zip file that contains** this plugin or None if it is a builtin plugin
- **self.site_customization: Stores a customization string entered** by the user.

Methods that should be overridden in sub classes:

- *initialize()* (stránka 195)
- *customization_help()* (stránka 195)

Useful methods:

- *temporary_file()* (stránka 195)
- __enter__()
- *load_resources()* (stránka 195)

supported_platforms = []

List of platforms this plugin works on. For example: ['windows', 'osx', 'linux']

name = 'Trivial Plugin'

The name of this plugin. You must set it something other than Trivial Plugin for it to work.

version = (1, 0, 0)

The version of this plugin as a 3-tuple (major, minor, revision)

description = u'Ned\u011bl\xe1 v\u016fbec nic'

A short string describing what this plugin does

author = u'Nezn\xe1m\xfd'

The author of this plugin

priority = 1

When more than one plugin exists for a filetype, the plugins are run in order of decreasing priority i.e. plugins with higher priority will be run first. The highest possible priority is sys.maxint. Default priority is 1.

minimum_calibre_version = (0, 4, 118)

The earliest version of calibre this plugin requires

can_be_disabled = True

If False, the user will not be able to disable this plugin. Use with care.

type = u'Z\xe1klad'
The type of this plugin. Used for categorizing plugins in the GUI

initialize()
Called once when calibre plugins are initialized. Plugins are re-initialized every time a new plugin is added. Also note that if the plugin is run in a worker process, such as for adding books, then the plugin will be initialized for every new worker process.

Perform any plugin specific initialization here, such as extracting resources from the plugin zip file. The path to the zip file is available as `self.plugin_path`.

Note that `self.site_customization` is **not** available at this point.

config_widget()
Implement this method and *save_settings()* (stránka 195) in your plugin to use a custom configuration dialog, rather then relying on the simple string based default customization.

This method, if implemented, must return a QWidget. The widget can have an optional method validate() that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns True.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

save_settings(*config_widget*)
Save the settings specified by the user with config_widget.

Parametry **config_widget** – The widget returned by *config_widget()* (stránka 195).

do_user_config(*parent=None*)
This method shows a configuration dialog for this plugin. It returns True if the user clicks OK, False otherwise. The changes are automatically applied.

load_resources(*names*)
If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

For example to load an image:

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png']).itervalues().next())
icon = QIcon(pixmap)
```

Parametry **names** – List of paths to resources in the zip file using / as separator

Vrací A dictionary of the form `{name: file_contents}`. Any names that were not found in the zip file will not be present in the dictionary.

customization_help(*gui=False*)
Return a string giving help on how to customize this plugin. By default raise a `NotImplementedError`, which indicates that the plugin does not require customization.

If you re-implement this method in your subclass, the user will be asked to enter a string as customization for this plugin. The customization string will be available as `self.site_customization`.

Site customization could be anything, for example, the path to a needed binary on the user's computer.

Parametry **gui** – If True return HTML help, otherwise return plain text help.

temporary_file(*suffix*)
Return a file-like object that is a temporary file on the file system. This file will remain available even after

being closed and will only be removed on interpreter shutdown. Use the `name` member of the returned object to access the full path to the created temporary file.

Parametry **suffix** – The suffix that the temporary file will have.

cli_main(*args*)
This method is the main entry point for your plugins command line interface. It is called when the user does: calibre-debug -r "Plugin Name". Any arguments passed are present in the args variable.

FileTypePlugin

class calibre.customize.**FileTypePlugin**(*plugin_path*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

A plugin that is associated with a particular set of file types.

file_types = set([])
Set of file types for which this plugin should be run. For example: `{'lit', 'mobi', 'prc'}`

on_import = False
If True, this plugin is run when books are added to the database

on_postimport = False
If True, this plugin is run after books are added to the database

on_preprocess = False
If True, this plugin is run just before a conversion

on_postprocess = False
If True, this plugin is run after conversion on the final file produced by the conversion output plugin.

run(*path_to_ebook*)
Run the plugin. Must be implemented in subclasses. It should perform whatever modifications are required on the ebook and return the absolute path to the modified ebook. If no modifications are needed, it should return the path to the original ebook. If an error is encountered it should raise an Exception. The default implementation simply return the path to the original ebook.

The modified ebook file should be created with the `temporary_file()` method.

Parametry **path_to_ebook** – Absolute path to the ebook.

Vrací Absolute path to the modified ebook.

postimport(*book_id*, *book_format*, *db*)
Called post import, i.e., after the book file has been added to the database.

Parametry

- **book_id** – Database id of the added book.
- **book_format** – The file type of the book that was added.
- **db** – Library database.

Metadata plugins

class calibre.customize.**MetadataReaderPlugin**(**args*, ***kwargs*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

A plugin that implements reading metadata from a set of file types.

file_types = set([])
Set of file types for which this plugin should be run For example: `set(['lit', 'mobi', 'prc'])`

get_metadata(*stream*, *type*)
Return metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data. :param type: The type of file. Guaranteed to be one of the entries in *file_types* (stránka 196). :return: A `calibre.ebooks.metadata.book.Metadata` object

class `calibre.customize.`**MetadataWriterPlugin**(**args*, ***kwargs*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

A plugin that implements reading metadata from a set of file types.

file_types = set([])
Set of file types for which this plugin should be run For example: `set(['lit', 'mobi', 'prc'])`

set_metadata(*stream*, *mi*, *type*)
Set metadata for the file represented by stream (a file like object that supports reading). Raise an exception when there is an error with the input data. :param type: The type of file. Guaranteed to be one of the entries in *file_types* (stránka 197). :param mi: A `calibre.ebooks.metadata.book.Metadata` object

Catalog plugins

class `calibre.customize.`**CatalogPlugin**(*plugin_path*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

A plugin that implements a catalog generator.

file_types = set([])
Output file type for which this plugin should be run For example: 'epub' or 'xml'

cli_options = []
CLI parser options specific to this plugin, declared as namedtuple Option:

```
from collections import namedtuple
Option = namedtuple('Option', 'option, default, dest, help')
cli_options = [Option('--catalog-title',
                      default = 'My Catalog',
                      dest = 'catalog_title',
                      help = (_('Title of generated catalog. \nDefault:') + " '" +
                      '%default' + "'"))]
cli_options parsed in library.cli:catalog_option_parser()
```

initialize()
If plugin is not a built-in, copy the plugin's .ui and .py files from the zip file to $TMPDIR. Tab will be dynamically generated and added to the Catalog Options dialog in calibre.gui2.dialogs.catalog.py:Catalog

run(*path_to_output*, *opts*, *db*, *ids*, *notification=None*)
Run the plugin. Must be implemented in subclasses. It should generate the catalog in the format specified in file_types, returning the absolute path to the generated catalog file. If an error is encountered it should raise an Exception.

The generated catalog file should be created with the `temporary_file()` method.

Parametry

- **path_to_output** – Absolute path to the generated catalog file.
- **opts** – A dictionary of keyword arguments

- **db** – A LibraryDatabase2 object

Metadata download plugins

class calibre.ebooks.metadata.sources.base.**Source**(**args*, ***kwargs*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

capabilities = frozenset([])
Set of capabilities supported by this plugin. Useful capabilities are: 'identify', 'cover'

touched_fields = frozenset([])
List of metadata fields that can potentially be download by this plugin during the identify phase

has_html_comments = False
Set this to True if your plugin returns HTML formatted comments

supports_gzip_transfer_encoding = False
Setting this to True means that the browser object will add Accept-Encoding: gzip to all requests. This can speedup downloads but make sure that the source actually supports gzip transfer encoding correctly first

cached_cover_url_is_reliable = True
Cached cover URLs can sometimes be unreliable (i.e. the download could fail or the returned image could be bogus. If that is often the case with this source set to False

options = ()
A list of Option objects. They will be used to automatically construct the configuration widget for this plugin

config_help_message = None
A string that is displayed at the top of the config widget for this plugin

can_get_multiple_covers = False
If True this source can return multiple covers for a given query

auto_trim_covers = False
If set to True covers downloaded by this plugin are automatically trimmed.

prefer_results_with_isbn = True
If set to True, and this source returns multiple results for a query, some of which have ISBNs and some of which do not, the results without ISBNs will be ignored

is_configured()
Return False if your plugin needs to be configured before it can be used. For example, it might need a username/password/API key.

get_author_tokens(*authors*, *only_first_author=True*)
Take a list of authors and return a list of tokens useful for an AND search query. This function tries to return tokens in first name middle names last name order, by assuming that if a comma is in the author name, the name is in lastname, other names form.

get_title_tokens(*title*, *strip_joiners=True*, *strip_subtitle=False*)
Take a title and return a list of tokens useful for an AND search query. Excludes connectives(optionally) and punctuation.

split_jobs(*jobs*, *num*)
Split a list of jobs into at most num groups, as evenly as possible

test_fields(*mi*)
Return the first field from self.touched_fields that is null on the mi object

clean_downloaded_metadata(*mi*)
Call this method in your plugin's identify method to normalize metadata before putting the Metadata object into result_queue. You can of course, use a custom algorithm suited to your metadata source.

get_book_url(*identifiers*)
Return a 3-tuple or None. The 3-tuple is of the form: (identifier_type, identifier_value, URL). The URL is the URL for the book identified by identifiers at this source. identifier_type, identifier_value specify the identifier corresponding to the URL. This URL must be browseable to by a human using a browser. It is meant to provide a clickable link for the user to easily visit the books page at this source. If no URL is found, return None. This method must be quick, and consistent, so only implement it if it is possible to construct the URL from a known scheme given identifiers.

get_book_url_name(*idtype*, *idval*, *url*)
Return a human readable name from the return value of get_book_url().

get_book_urls(*identifiers*)
Override this method if you would like to return multiple urls for this book. Return a list of 3-tuples. By default this method simply calls `get_book_url()` (stránka 199).

get_cached_cover_url(*identifiers*)
Return cached cover URL for the book identified by the identifiers dict or None if no such URL exists.

Note that this method must only return validated URLs, i.e. not URLS that could result in a generic cover image or a not found error.

identify_results_keygen(*title=None*, *authors=None*, *identifiers={}*)
Return a function that is used to generate a key that can sort Metadata objects by their relevance given a search query (title, authors, identifiers).

These keys are used to sort the results of a call to `identify()` (stránka 199).

For details on the default algorithm see `InternalMetadataCompareKeyGen` (stránka 200). Reimplement this function in your plugin if the default algorithm is not suitable.

identify(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*)
Identify a book by its title/author/isbn/etc.

If identifiers(s) are specified and no match is found and this metadata source does not store all related identifiers (for example, all ISBNs of a book), this method should retry with just the title and author (assuming they were specified).

If this metadata source also provides covers, the URL to the cover should be cached so that a subsequent call to the get covers API with the same ISBN/special identifier does not need to get the cover URL again. Use the caching API for this.

Every Metadata object put into result_queue by this method must have a *source_relevance* attribute that is an integer indicating the order in which the results were returned by the metadata source for this query. This integer will be used by `compare_identify_results()`. If the order is unimportant, set it to zero for every result.

Make sure that any cover/isbn mapping information is cached before the Metadata object is put into result_queue.

Parametry

- **log** – A log object, use it to output debugging information/errors
- **result_queue** – A result Queue, results should be put into it. Each result is a Metadata object
- **abort** – If abort.is_set() returns True, abort further processing and return as soon as possible

- **title** – The title of the book, can be None
- **authors** – A list of authors of the book, can be None
- **identifiers** – A dictionary of other identifiers, most commonly {'isbn':'1234...'}
- **timeout** – Timeout in seconds, no network request should hang for longer than timeout.

Vrací None if no errors occurred, otherwise a unicode representation of the error suitable for showing to the user

download_cover(*log*, *result_queue*, *abort*, *title=None*, *authors=None*, *identifiers={}*, *timeout=30*, *get_best_cover=False*)

Download a cover and put it into result_queue. The parameters all have the same meaning as for *identify()* (stránka 199). Put (self, cover_data) into result_queue.

This method should use cached cover URLs for efficiency whenever possible. When cached data is not present, most plugins simply call identify and use its results.

If the parameter get_best_cover is True and this plugin can get multiple covers, it should only get the "best" one.

class calibre.ebooks.metadata.sources.base.**InternalMetadataCompareKeyGen**(*mi*, *source_plugin*, *title*, *authors*, *identifiers*)

Generate a sort key for comparison of the relevance of Metadata objects, given a search query. This is used only to compare results from the same metadata source, not across different sources.

The sort key ensures that an ascending order sort is a sort by order of decreasing relevance.

The algorithm is:

•Prefer results that have at least one identifier the same as for the query

•Prefer results with a cached cover URL

•Prefer results with all available fields filled in

•Prefer results with the same language as the current user interface language

•Prefer results that are an exact title match to the query

•Prefer results with longer comments (greater than 10% longer)

•**Use the relevance of the result as reported by the metadata source's search** engine

Conversion plugins

class calibre.customize.conversion.**InputFormatPlugin**(**args*)

Nadtřídy: *calibre.customize.Plugin* (stránka 194)

InputFormatPlugins are responsible for converting a document into HTML+OPF+CSS+etc. The results of the conversion *must* be encoded in UTF-8. The main action happens in *convert()* (stránka 201).

file_types = set([])

Set of file types for which this plugin should be run For example: `set(['azw', 'mobi', 'prc'])`

is_image_collection = False
If True, this input plugin generates a collection of images, one per HTML file. This can be set dynamically, in the convert method if the input files can be both image collections and non-image collections. If you set this to True, you must implement the get_images() method that returns a list of images.

core_usage = 1
Number of CPU cores used by this plugin A value of -1 means that it uses all available cores

for_viewer = False
If set to True, the input plugin will perform special processing to make its output suitable for viewing

output_encoding = 'utf-8'
The encoding that this input plugin creates files in. A value of None means that the encoding is undefined and must be detected individually

common_options = set([<calibre.customize.conversion.OptionRecommendation object at 0x7f137193c6d0>])
Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (stránka 201) instead. Every option must be an instance of `OptionRecommendation`.

options = set([])
Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = set([])
A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

get_images()
Return a list of absolute paths to the images, if this input plugin represents an image collection. The list of images is in the same order as the spine and the TOC.

convert(*stream*, *options*, *file_ext*, *log*, *accelerators*)
This method must be implemented in sub-classes. It must return the path to the created OPF file or an `OEBBook` instance. All output should be contained in the current directory. If this plugin creates files outside the current directory they must be deleted/marked for deletion before this method returns.

Parametry

- **stream** – A file like object that contains the input file.
- **options** – Options to customize the conversion process. Guaranteed to have attributes corresponding to all the options declared by this plugin. In addition, it will have a verbose attribute that takes integral values from zero upwards. Higher numbers mean be more verbose. Another useful attribute is `input_profile` that is an instance of `calibre.customize.profiles.InputProfile`.
- **file_ext** – The extension (without the .) of the input file. It is guaranteed to be one of the *file_types* supported by this plugin.
- **log** – A `calibre.utils.logging.Log` object. All output should use this object.
- **accelarators** – A dictionary of various information that the input plugin can get easily that would speed up the subsequent stages of the conversion.

postprocess_book(*oeb*, *opts*, *log*)
Called to allow the input plugin to perform postprocessing after the book has been parsed.

specialize(*oeb*, *opts*, *log*, *output_fmt*)
Called to allow the input plugin to specialize the parsed book for a particular output format. Called after postprocess_book and before any transforms are performed on the parsed book.

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)
Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin input plugins for examples.

class calibre.customize.conversion.**OutputFormatPlugin**(**args*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

OutputFormatPlugins are responsible for converting an OEB document (OPF+HTML) into an output ebook.

The OEB document can be assumed to be encoded in UTF-8. The main action happens in *convert()* (stránka 202).

file_type = None
The file type (extension without leading period) that this plugin outputs

common_options = set([<calibre.customize.conversion.OptionRecommendation object at 0x7f137193c850>])
Options shared by all Input format plugins. Do not override in sub-classes. Use *options* (stránka 202) instead. Every option must be an instance of `OptionRecommendation`.

options = set([])
Options to customize the behavior of this plugin. Every option must be an instance of `OptionRecommendation`.

recommendations = set([])
A set of 3-tuples of the form (option_name, recommended_value, recommendation_level)

convert(*oeb_book*, *output*, *input_plugin*, *opts*, *log*)
Render the contents of *oeb_book* (which is an instance of `calibre.ebooks.oeb.OEBBook` to the file specified by output.

Parametry

- **output** – Either a file like object or a string. If it is a string it is the path to a directory that may or may not exist. The output plugin should write its output into that directory. If it is a file like object, the output plugin should write its output into the file.
- **input_plugin** – The input plugin that was used at the beginning of the conversion pipeline.
- **opts** – Conversion options. Guaranteed to have attributes corresponding to the OptionRecommendations of this plugin.
- **log** – The logger. Print debug/info messages etc. using this.

specialize_css_for_output(*log*, *opts*, *item*, *stylizer*)
Can be used to make changes to the css during the CSS flattening process.

Parametry

- **item** – The item (HTML file) being processed
- **stylizer** – A Stylizer object containing the flattened styles for item. You can get the style for any element by stylizer.style(element).

gui_configuration_widget(*parent*, *get_option_by_name*, *get_option_help*, *db*, *book_id=None*)
Called to create the widget used for configuring this plugin in the calibre GUI. The widget must be an instance of the PluginWidget class. See the builtin output plugins for examples.

Device Drivers

The base class for all device drivers is *DevicePlugin* (stránka 203). However, if your device exposes itself as a USBMS drive to the operating system, you should use the USBMS class instead as it implements all the logic needed

to support these kinds of devices.

class calibre.devices.interface.**DevicePlugin**(*plugin_path*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

Defines the interface that should be implemented by backends that communicate with an ebook reader.

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']
Ordered list of supported formats

VENDOR_ID = 0
VENDOR_ID can be either an integer, a list of integers or a dictionary If it is a dictionary, it must be a dictionary of dictionaries, of the form:

```
{
 integer_vendor_id : { product_id : [list of BCDs], ... },
 ...
}
```

PRODUCT_ID = 0
An integer or a list of integers

BCD = None
BCD can be either None to not distinguish between devices based on BCD, or it can be a list of the BCD numbers of all devices supported by this driver.

THUMBNAIL_HEIGHT = 68
Height for thumbnails on the device

THUMBNAIL_COMPRESSION_QUALITY = 75
Width for thumbnails on the device. Setting this will force thumbnails to this size, not preserving aspect ratio. If it is not set, then the aspect ratio will be preserved and the thumbnail will be no higher than THUMBNAIL_HEIGHT Compression quality for thumbnails. Set this closer to 100 to have better quality thumbnails with fewer compression artifacts. Of course, the thumbnails get larger as well.

WANTS_UPDATED_THUMBNAILS = False
Set this to True if the device supports updating cover thumbnails during sync_booklists. Setting it to true will ask device.py to refresh the cover thumbnails during book matching

CAN_SET_METADATA = ['title', 'authors', 'collections']
Whether the metadata on books can be set via the GUI.

CAN_DO_DEVICE_DB_PLUGBOARD = False
Whether the device can handle device_db metadata plugboards

path_sep = '/'
Path separator for paths to books on device

icon = '/home/kovid/work/calibre/resources/images/reader.png'
Icon for this device

UserAnnotation
alias třídy Annotation

OPEN_FEEDBACK_MESSAGE = None
GUI displays this as a message if not None. Useful if opening can take a long time

VIRTUAL_BOOK_EXTENSIONS = frozenset([])
Set of extensions that are "virtual books" on the device and therefore cannot be viewed/saved/added to library. For example: frozenset(['kobo'])

VIRTUAL_BOOK_EXTENSION_MESSAGE = None
Message to display to user for virtual book extensions.

NUKE_COMMENTS = None
Whether to nuke comments in the copy of the book sent to the device. If not None this should be short string that the comments will be replaced by.

MANAGES_DEVICE_PRESENCE = False
If True indicates that this driver completely manages device detection, ejecting and so forth. If you set this to True, you *must* implement the detect_managed_devices and debug_managed_device_detection methods. A driver with this set to true is responsible for detection of devices, managing a blacklist of devices, a list of ejected devices and so forth. calibre will periodically call the detect_managed_devices() method and if it returns a detected device, calibre will call open(). open() will be called every time a device is returned even is previous calls to open() failed, therefore the driver must maintain its own blacklist of failed devices. Similarly, when ejecting, calibre will call eject() and then assuming the next call to detect_managed_devices() returns None, it will call post_yank_cleanup().

SLOW_DRIVEINFO = False
If set the True, calibre will call the *get_driveinfo()* (stránka 206) method after the books lists have been loaded to get the driveinfo.

ASK_TO_ALLOW_CONNECT = False
If set to True, calibre will ask the user if they want to manage the device with calibre, the first time it is detected. If you set this to True you must implement *get_device_uid()* (stránka 208) and *ignore_connected_device()* (stránka 208) and *get_user_blacklisted_devices()* (stránka 208) and *set_user_blacklisted_devices()* (stránka 208)

user_feedback_after_callback = None
Set this to a dictionary of the form {'title':title, 'msg':msg, 'det_msg':detailed_msg} to have calibre popup a message to the user after some callbacks are run (currently only upload_books). Be careful to not spam the user with too many messages. This variable is checked after *every* callback, so only set it when you really need to.

is_usb_connected(*devices_on_system*, *debug=False*, *only_presence=False*)
Return True, device_info if a device handled by this plugin is currently connected.

Parametry **devices_on_system** – List of devices currently connected

detect_managed_devices(*devices_on_system*, *force_refresh=False*)
Called only if MANAGES_DEVICE_PRESENCE is True.

Scan for devices that this driver can handle. Should return a device object if a device is found. This object will be passed to the open() method as the connected_device. If no device is found, return None. The returned object can be anything, calibre does not use it, it is only passed to open().

This method is called periodically by the GUI, so make sure it is not too resource intensive. Use a cache to avoid repeatedly scanning the system.

Parametry

- **devices_on_system** – Set of USB devices found on the system.
- **force_refresh** – If True and the driver uses a cache to prevent repeated scanning, the cache must be flushed.

debug_managed_device_detection(*devices_on_system*, *output*)
Called only if MANAGES_DEVICE_PRESENCE is True.

Should write information about the devices detected on the system to output, which is a file like object.

Should return True if a device was detected and successfully opened, otherwise False.

reset(*key='-1'*, *log_packets=False*, *report_progress=None*, *detected_device=None*)

Parametry

- **key** – The key to unlock the device
- **log_packets** – If true the packet stream to/from the device is logged
- **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks If it is called with -1 that means that the task does not have any progress information
- **detected_device** – Device information from the device scanner

can_handle_windows(*device_id*, *debug=False*)
Optional method to perform further checks on a device to see if this driver is capable of handling it. If it is not it should return False. This method is only called after the vendor, product ids and the bcd have matched, so it can do some relatively time intensive checks. The default implementation returns True. This method is called only on windows. See also *can_handle()* (stránka 205).

Parametry **device_info** – On windows a device ID string. On Unix a tuple of `(vendor_id, product_id, bcd)`.

can_handle(*device_info*, *debug=False*)
Unix version of *can_handle_windows()* (stránka 205)

Parametry **device_info** – Is a tuple of (vid, pid, bcd, manufacturer, product, serial number)

open(*connected_device*, *library_uuid*)
Perform any device specific initialization. Called after the device is detected but before any other functions that communicate with the device. For example: For devices that present themselves as USB Mass storage devices, this method would be responsible for mounting the device or if the device has been automounted, for finding out where it has been mounted. The method `calibre.devices.usbms.device.Device.open()` has an implementation of this function that should serve as a good example for USB Mass storage devices.

This method can raise an OpenFeedback exception to display a message to the user.

Parametry

- **connected_device** – The device that we are trying to open. It is a tuple of (vendor id, product id, bcd, manufacturer name, product name, device serial number). However, some devices have no serial number and on windows only the first three fields are present, the rest are None.
- **library_uuid** – The UUID of the current calibre library. Can be None if there is no library (for example when used from the command line).

eject()
Un-mount / eject the device from the OS. This does not check if there are pending GUI jobs that need to communicate with the device.

NOTE: That this method may not be called on the same thread as the rest of the device methods.

post_yank_cleanup()
Called if the user yanks the device without ejecting it first.

set_progress_reporter(*report_progress*)
Set a function to report progress information.

Parametry **report_progress** – Function that is called with a % progress (number between 0 and 100) for various tasks If it is called with -1 that means that the task does not have any progress information

get_device_information(*end_session=True*)
Ask device for device information. See L{DeviceInfoQuery}.

Vrací (device name, device version, software version on device, mime type) The tuple can optionally have a fifth element, which is a drive information dictionary. See usbms.driver for an example.

get_driveinfo()
Return the driveinfo dictionary. Usually called from get_device_information(), but if loading the driveinfo is slow for this driver, then it should set SLOW_DRIVEINFO. In this case, this method will be called by calibre after the book lists have been loaded. Note that it is not called on the device thread, so the driver should cache the drive info in the books() method and this function should return the cached data.

card_prefix(*end_session=True*)
Return a 2 element list of the prefix to paths on the cards. If no card is present None is set for the card's prefix. E.G. ('/place', '/place2') (None, 'place2') ('place', None) (None, None)

total_space(*end_session=True*)

Get total space available on the mountpoints:

1. Hlavní paměť
2. Memory Card A
3. Memory Card B

Vrací A 3 element list with total space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return 0.

free_space(*end_session=True*)

Get free space available on the mountpoints:

1. Hlavní paměť
2. Karta A
3. Karta B

Vrací A 3 element list with free space in bytes of (1, 2, 3). If a particular device doesn't have any of these locations it should return -1.

books(*oncard=None*, *end_session=True*)
Return a list of ebooks on the device.

Parametry **oncard** – If 'carda' or 'cardb' return a list of ebooks on the specific storage card, otherwise return list of ebooks in main memory of device. If a card is specified and no books are on the card return empty list.

Vrací A BookList.

upload_books(*files*, *names*, *on_card=None*, *end_session=True*, *metadata=None*)
Upload a list of books to the device. If a file already exists on the device, it should be replaced. This method should raise a `FreeSpaceError` if there is not enough free space on the device. The text of the FreeSpaceError must contain the word "card" if `on_card` is not None otherwise it must contain the word "memory".

Parametry

- **files** – A list of paths
- **names** – A list of file names that the books should have once uploaded to the device. len(names) == len(files)

- **metadata** – If not None, it is a list of `Metadata` objects. The idea is to use the metadata to determine where on the device to put the book. len(metadata) == len(files). Apart from the regular cover (path to cover), there may also be a thumbnail attribute, which should be used in preference. The thumbnail attribute is of the form (width, height, cover_data as jpeg).

Vrací A list of 3-element tuples. The list is meant to be passed to *add_books_to_metadata()* (stránka 207).

classmethod add_books_to_metadata(*locations*, *metadata*, *booklists*)
Add locations to the booklists. This function must not communicate with the device.

Parametry

- **locations** – Result of a call to L{upload_books}
- **metadata** – List of `Metadata` objects, same as for *upload_books()* (stránka 206).
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth'books(oncard='cardb')').

delete_books(*paths*, *end_session=True*)
Delete books at paths on device.

classmethod remove_books_from_metadata(*paths*, *booklists*)
Remove books from the metadata list. This function must not communicate with the device.

Parametry

- **paths** – paths to books on the device.
- **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth'books(oncard='cardb')').

sync_booklists(*booklists*, *end_session=True*)
Update metadata on device.

Parametry **booklists** – A tuple containing the result of calls to (`books(oncard=None)()`, `books(oncard='carda')()`, :meth'books(oncard='cardb')').

get_file(*path*, *outfile*, *end_session=True*)
Read the file at `path` on the device and write it to outfile.

Parametry **outfile** – file object like `sys.stdout` or the result of an *open()* (stránka 205) call.

classmethod config_widget()
Should return a QWidget. The QWidget contains the settings for the device interface

classmethod save_settings(*settings_widget*)
Should save settings to disk. Takes the widget created in *config_widget()* (stránka 207) and saves all settings to disk.

classmethod settings()
Should return an opts object. The opts object should have at least one attribute *format_map* which is an ordered list of formats for the device.

set_plugboards(*plugboards*, *pb_func*)
provide the driver the current set of plugboards and a function to select a specific plugboard. This method is called immediately before add_books and sync_booklists.

pb_func is a callable with the following signature:: def pb_func(device_name, format, plugboards)

You give it the current device name (either the class name or DEVICE_PLUGBOARD_NAME), the format you are interested in (a 'real' format or 'device_db'), and the plugboards (you were given those by set_plugboards, the same place you got this method).

Vrací None or a single plugboard instance.

set_driveinfo_name(*location_code*, *name*)
Set the device name in the driveinfo file to 'name'. This setting will persist until the file is re-created or the name is changed again.

Non-disk devices should implement this method based on the location codes returned by the get_device_information() method.

prepare_addable_books(*paths*)
Given a list of paths, returns another list of paths. These paths point to addable versions of the books.

If there is an error preparing a book, then instead of a path, the position in the returned list for that book should be a three tuple: (original_path, the exception instance, traceback)

startup()
Called when calibre is is starting the device. Do any initialization required. Note that multiple instances of the class can be instantiated, and thus __init__ can be called multiple times, but only one instance will have this method called. This method is called on the device thread, not the GUI thread.

shutdown()
Called when calibre is shutting down, either for good or in preparation to restart. Do any cleanup required. This method is called on the device thread, not the GUI thread.

get_device_uid()
Must return a unique id for the currently connected device (this is called immediately after a successful call to open()). You must implement this method if you set ASK_TO_ALLOW_CONNECT = True

ignore_connected_device(*uid*)
Should ignore the device identified by uid (the result of a call to get_device_uid()) in the future. You must implement this method if you set ASK_TO_ALLOW_CONNECT = True. Note that this function is called immediately after open(), so if open() caches some state, the driver should reset that state.

get_user_blacklisted_devices()
Return map of device uid to friendly name for all devices that the user has asked to be ignored.

set_user_blacklisted_devices(*devices*)
Set the list of device uids that should be ignored by this driver.

specialize_global_preferences(*device_prefs*)
Implement this method if your device wants to override a particular preference. You must ensure that all call sites that want a preference that can be overridden use device_prefs['something'] instead of prefs['something']. Your method should call device_prefs.set_overrides(pref=val, pref=val, ...). Currently used for: metadata management (prefs['manage_device_metadata'])

set_library_info(*library_name*, *library_uuid*, *field_metadata*)
Implement this method if you want information about the current calibre library. This method is called at startup and when the calibre library changes while connected.

is_dynamically_controllable()
Called by the device manager when starting plugins. If this method returns a string, then a) it supports the device manager's dynamic control interface, and b) that name is to be used when talking to the plugin.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

start_plugin()
This method is called to start the plugin. The plugin should begin to accept device connections however it does that. If the plugin is already accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

stop_plugin()
This method is called to stop the plugin. The plugin should no longer accept connections, and should cleanup behind itself. It is likely that this method should call shutdown. If the plugin is already not accepting connections, then do nothing.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

get_option(*opt_string*, *default=None*)
Return the value of the option indicated by opt_string. This method can be called when the plugin is not started. Return None if the option does not exist.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

set_option(*opt_string*, *opt_value*)
Set the value of the option indicated by opt_string. This method can be called when the plugin is not started.

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

is_running()
Return True if the plugin is started, otherwise false

This method can be called on the GUI thread. A driver that implements this method must be thread safe.

synchronize_with_db(*db*, *book_id*, *book_metadata*, *first_call*)
Called during book matching when a book on the device is matched with a book in calibre's db. The method is responsible for syncronizing data from the device to calibre's db (if needed).

The method must return a two-value tuple. The first value is a set of calibre book ids changed if calibre's database was changed or None if the database was not changed. If the first value is an empty set then the metadata for the book on the device is updated with calibre's metadata and given back to the device, but no GUI refresh of that book is done. This is useful when the calibre data is correct but must be sent to the device.

The second value is itself a 2-value tuple. The first value in the tuple specifies whether a book format should be sent to the device. The intent is to permit verifying that the book on the device is the same as the book in calibre. This value must be None if no book is to be sent, otherwise return the base file name on the device (a string like foobar.epub). Be sure to include the extension in the name. The device subsystem will construct a send_books job for all books with not- None returned values. Note: other than to later retrieve the extension, the name is ignored in cases where the device uses a template to generate the file name, which most do. The second value in the returned tuple indicated whether the format is future-dated. Return True if it is, otherwise return False. Calibre will display a dialog to the user listing all future dated books.

Extremely important: this method is called on the GUI thread. It must be threadsafe with respect to the device manager's thread.

book_id: the calibre id for the book in the database. book_metadata: the Metadata object for the book coming from the device. first_call: True if this is the first call during a sync, False otherwise

class calibre.devices.interface.**BookList**(*oncard*, *prefix*, *settings*)
Nadtřídy: list

A list of books. Each Book object must have the fields

1.title

2.authors

3.size (file size of the book)

4.datetime (a UTC time tuple)

5.path (path on the device to the book)

6.thumbnail (can be None) thumbnail is either a str/bytes object with the image data or it should have an attribute image_path that stores an absolute (platform native) path to the image

7.tags (a list of strings, can be empty).

supports_collections()
Return True if the device supports collections for this book list.

add_book(*book*, *replace_metadata*)
Add the book to the booklist. Intent is to maintain any device-internal metadata. Return True if booklists must be sync'ed

remove_book(*book*)
Remove a book from the booklist. Correct any device metadata at the same time

get_collections(*collection_attributes*)
Return a dictionary of collections created from collection_attributes. Each entry in the dictionary is of the form collection name:[list of books]

The list of books is sorted by book title, except for collections created from series, in which case series_index is used.

Parametry **collection_attributes** – A list of attributes of the Book object

USB Mass Storage based devices

The base class for such devices is *calibre.devices.usbms.driver.USBMS* (stránka 211). This class in turn inherits some of its functionality from its bases, documented below. A typical basic USBMS based driver looks like this:

```
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
    gui_name = 'PD Novel'
    description = _('Communicate with the Pandigital Novel')
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'linux', 'osx']
    FORMATS = ['epub', 'pdf']

    VENDOR_ID   = [0x18d1]
    PRODUCT_ID  = [0xb004]
    BCD         = [0x224]

    VENDOR_NAME = 'ANDROID'
    WINDOWS_MAIN_MEM = WINDOWS_CARD_A_MEM = '__UMS_COMPOSITE'
    THUMBNAIL_HEIGHT = 144

    EBOOK_DIR_MAIN = 'eBooks'
    SUPPORTS_SUB_DIRS = False

    def upload_cover(self, path, filename, metadata):
        coverdata = getattr(metadata, 'thumbnail', None)
        if coverdata and coverdata[2]:
            with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
                coverfile.write(coverdata[2])
```

class calibre.devices.usbms.device.**Device**(*plugin_path*)
Nadtřídy: calibre.devices.usbms.deviceconfig.DeviceConfig, *calibre.devices.interface.DevicePlugin* (stránka 203)

This class provides logic common to all drivers for devices that export themselves as USB Mass Storage devices. Provides implementations for mounting/ejecting of USBMS devices on all platforms.

WINDOWS_MAIN_MEM = None
String identifying the main memory of the device in the windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_A_MEM = None
String identifying the first card of the device in the windows PnP id strings This can be None, string, list of strings or compiled regex

WINDOWS_CARD_B_MEM = None
String identifying the second card of the device in the windows PnP id strings This can be None, string, list of strings or compiled regex

OSX_MAIN_MEM_VOL_PAT = None
Used by the new driver detection to disambiguate main memory from storage cards. Should be a regular expression that matches the main memory mount point assigned by OS X

MAX_PATH_LEN = 250
The maximum length of paths created on the device

NEWS_IN_FOLDER = True
Put news in its own folder

windows_sort_drives(*drives*)
Called to disambiguate main memory and storage card for devices that do not distinguish between them on the basis of *WINDOWS_CARD_NAME*. For e.g.: The EB600

sanitize_callback(*path*)
Callback to allow individual device drivers to override the path sanitization used by create_upload_path().

filename_callback(*default*, *mi*)
Callback to allow drivers to change the default file name set by create_upload_path().

sanitize_path_components(*components*)
Perform any device specific sanitization on the path components for files to be uploaded to the device

get_annotations(*path_map*)
Resolve path_map to annotation_map of files found on the device

add_annotation_to_library(*db*, *db_id*, *annotation*)
Add an annotation to the calibre library

class calibre.devices.usbms.cli.**CLI**

class calibre.devices.usbms.driver.**USBMS**(*plugin_path*)
Nadtřídy: *calibre.devices.usbms.cli.CLI* (stránka 211), *calibre.devices.usbms.device.Device* (stránka 210)

The base class for all USBMS devices. Implements the logic for sending/getting/updating metadata/caching metadata/etc.

upload_cover(*path*, *filename*, *metadata*, *filepath*)
Upload book cover to the device. Default implementation does nothing.

Parametry

- **path** – The full path to the directory where the associated book is located.
- **filename** – The name of the book file without the extension.
- **metadata** – metadata belonging to the book. Use metadata.thumbnail for cover
- **filepath** – The full path to the ebook file

classmethod normalize_path(*path*)
Return path with platform native path separators

User Interface Actions

If you are adding your own plugin in a zip file, you should subclass both InterfaceActionBase and InterfaceAction. The `load_actual_plugin()` method of you InterfaceActionBase subclass must return an instantiated object of your InterfaceBase subclass.

class `calibre.gui2.actions.`**InterfaceAction**(*parent*, *site_customization*)
Nadtřídy: `PyQt5.QtCore.QObject`

A plugin representing an "action" that can be taken in the graphical user interface. All the items in the toolbar and context menus are implemented by these plugins.

Note that this class is the base class for these plugins, however, to integrate the plugin with calibre's plugin system, you have to make a wrapper class that references the actual plugin. See the `calibre.customize.builtins` module for examples.

If two `InterfaceAction` objects have the same name, the one with higher priority takes precedence.

Sub-classes should implement the `genesis()`, `library_changed()`, `location_selected()` `shutting_down()` and `initialization_complete()` methods.

Once initialized, this plugin has access to the main calibre GUI via the `gui` member. You can access other plugins by name, for example:

```
self.gui.iactions['Save To Disk']
```

To access the actual plugin, use the `interface_action_base_plugin` attribute, this attribute only becomes available after the plugin has been initialized. Useful if you want to use methods from the plugin class like do_user_config().

The QAction specified by `action_spec` is automatically create and made available as `self.qaction`.

name = 'Implement me'
The plugin name. If two plugins with the same name are present, the one with higher priority takes precedence.

priority = 1
The plugin priority. If two plugins with the same name are present, the one with higher priority takes precedence.

popup_type = 1
The menu popup type for when this plugin is added to a toolbar

auto_repeat = False
Whether this action should be auto repeated when its shortcut key is held down.

action_spec = ('text', 'icon', None, None)
Of the form: (text, icon_path, tooltip, keyboard shortcut) icon, tooltip and keyboard shortcut can be None shortcut must be a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

action_add_menu = False
If True, a menu is automatically created and added to self.qaction

action_menu_clone_qaction = False
If True, a clone of self.qaction is added to the menu of self.qaction If you want the text of this action to be different from that of self.qaction, set this variable to the new text

dont_add_to = frozenset([])
Set of locations to which this action must not be added. See `all_locations` for a list of possible locations

dont_remove_from = frozenset([])
Set of locations from which this action must not be removed. See `all_locations` for a list of possible locations

action_type = 'global'
Type of action 'current' means acts on the current view 'global' means an action that does not act on the current view, but rather on calibre as a whole

accepts_drops = False
If True, then this InterfaceAction will have the opportunity to interact with drag and drop events. See the methods, `accept_enter_event()`, :meth':accept_drag_move_event', `drop_event()` for details.

accept_enter_event(*event*, *mime_data*)
This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

accept_drag_move_event(*event*, *mime_data*)
This method should return True iff this interface action is capable of handling the drag event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI.

drop_event(*event*, *mime_data*)
This method should perform some useful action and return True iff this interface action is capable of handling the drop event. Do not call accept/ignore on the event, that will be taken care of by the calibre UI. You should not perform blocking/long operations in this function. Instead emit a signal or use QTimer.singleShot and return quickly. See the builtin actions for examples.

create_menu_action(*menu*, *unique_name*, *text*, *icon=None*, *shortcut=None*, *description=None*, *triggered=None*, *shortcut_name=None*)
Convenience method to easily add actions to a QMenu. Returns the created QAction, This action has one extra attribute calibre_shortcut_unique_name which if not None refers to the unique name under which this action is registered with the keyboard manager.

Parametry

- **menu** – The QMenu the newly created action will be added to
- **unique_name** – A unique name for this action, this must be globally unique, so make it as descriptive as possible. If in doubt add a uuid to it.
- **text** – The text of the action.
- **icon** – Either a QIcon or a file name. The file name is passed to the I() builtin, so you do not need to pass the full path to the images directory.
- **shortcut** – A string, a list of strings, None or False. If False, no keyboard shortcut is registered for this action. If None, a keyboard shortcut with no default keybinding is registered. String and list of strings register a shortcut with default keybinding as specified.
- **description** – A description for this action. Used to set tooltips.
- **triggered** – A callable which is connected to the triggered signal of the created action.

- **shortcut_name** – The test displayed to the user when customizing the keyboard shortcuts for this action. By default it is set to the value of `text`.

load_resources(*names*)
If this plugin comes in a ZIP file (user added plugin), this method will allow you to load resources from the ZIP file.

For example to load an image:

```
pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png']).itervalues().next())
icon = QIcon(pixmap)
```

Parametry **names** – List of paths to resources in the zip file using / as separator

Vrací A dictionary of the form `{name : file_contents}`. Any names that were not found in the zip file will not be present in the dictionary.

genesis()
Setup this plugin. Only called once during initialization. self.gui is available. The action specified by `action_spec` is available as `self.qaction`.

location_selected(*loc*)
Called whenever the book list being displayed in calibre changes. Currently values for loc are: `library, main, card and cardb`.

This method should enable/disable this action and its sub actions as appropriate for the location.

library_changed(*db*)
Called whenever the current library is changed.

Parametry **db** – The LibraryDatabase corresponding to the current library.

gui_layout_complete()
Called once per action when the layout of the main GUI is completed. If your action needs to make changes to the layout, they should be done here, rather than in `initialization_complete()`.

initialization_complete()
Called once per action when the initialization of the main GUI is completed.

shutting_down()
Called once per plugin when the main GUI is in the process of shutting down. Release any used resources, but try not to block the shutdown for long periods of time.

Vrací False to halt the shutdown. You are responsible for telling the user why the shutdown was halted.

class calibre.customize.**InterfaceActionBase**(**args*, ***kwargs*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

load_actual_plugin(*gui*)
This method must return the actual interface action plugin object.

Preferences Plugins

class calibre.customize.**PreferencesPlugin**(*plugin_path*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

A plugin representing a widget displayed in the Preferences dialog.

This plugin has only one important method `create_widget()`. The various fields of the plugin control how it is categorized in the UI.

config_widget = None
Import path to module that contains a class named ConfigWidget which implements the ConfigWidgetInterface. Used by `create_widget()`.

category_order = 100
Where in the list of categories the `category` of this plugin should be.

name_order = 100
Where in the list of names in a category, the `gui_name` of this plugin should be

category = None
The category this plugin should be in

gui_category = None
The category name displayed to the user for this plugin

gui_name = None
The name displayed to the user for this plugin

icon = None
The icon for this plugin, should be an absolute path

description = None
The description used for tooltips and the like

create_widget(*parent=None*)
Create and return the actual Qt widget used for setting this group of preferences. The widget must implement the *calibre.gui2.preferences.ConfigWidgetInterface* (stránka 215).

The default implementation uses `config_widget` to instantiate the widget.

class calibre.gui2.preferences.**ConfigWidgetInterface**
This class defines the interface that all widgets displayed in the Preferences dialog must implement. See `ConfigWidgetBase` for a base class that implements this interface and defines various convenience methods as well.

changed_signal = None
This signal must be emitted whenever the user changes a value in this widget

supports_restoring_to_defaults = True
Set to True iff the `restore_to_defaults()` method is implemented.

restore_defaults_desc = u'Obnovit nastaven\xed na v\xfdchoz\xed hodnoty. Mus\xedte kliknout na Pou\u017e\xedt
The tooltip for the Restore to defaults button

restart_critical = False
If True the Preferences dialog will not allow the user to set any more preferences. Only has effect if `commit()` returns True.

genesis(*gui*)
Called once before the widget is displayed, should perform any necessary setup.

Parametry **gui** – The main calibre graphical user interface

initialize()
Should set all config values to their initial values (the values stored in the config files).

restore_defaults()
Should set all config values to their defaults.

commit()
Save any changed settings. Return True if the changes require a restart, False otherwise. Raise an `AbortCommit` exception to indicate that an error occurred. You are responsible for giving the user feedback about what the error is and how to correct it.

refresh_gui(*gui*)
Called once after this widget is committed. Responsible for causing the gui to reread any changed settings. Note that by default the GUI re-initializes various elements anyway, so most widgets won't need to use this method.

class `calibre.gui2.preferences.`**ConfigWidgetBase**(*parent=None*)
Base class that contains code to easily add standard config widgets like checkboxes, combo boxes, text fields and so on. See the `register()` method.

This class automatically handles change notification, resetting to default, translation between gui objects and config objects, etc. for registered settings.

If your config widget inherits from this class but includes setting that are not registered, you should override the `ConfigWidgetInterface` methods and call the base class methods inside the overrides.

register(*name*, *config_obj*, *gui_name=None*, *choices=None*, *restart_required=False*, *empty_string_is_None=True*, *setting=<class 'calibre.gui2.preferences.Setting'>*)
Register a setting.

Parametry

- **name** – The setting name
- **config** – The config object that reads/writes the setting
- **gui_name** – The name of the GUI object that presents an interface to change the setting. By default it is assumed to be `'opt_' + name`.
- **choices** – If this setting is a multiple choice (combobox) based setting, the list of choices. The list is a list of two element tuples of the form: `[(gui name, value), ...]`
- **setting** – The class responsible for managing this setting. The default class handles almost all cases, so this param is rarely used.

Viewer plugins

class `calibre.customize.`**ViewerPlugin**(*plugin_path*)
Nadtřídy: *calibre.customize.Plugin* (stránka 194)

type = u'Prohl\xed\u017ee\u010d'
These plugins are used to add functionality to the calibre viewer.

load_fonts()
This method is called once at viewer startup. It should load any fonts it wants to make available. For example:

```
def load_fonts():
    from PyQt5.Qt import QFontDatabase
    font_data = get_resources(['myfont1.ttf', 'myfont2.ttf'])
    for raw in font_data.itervalues():
        QFontDatabase.addApplicationFontFromData(raw)
```

load_javascript(*evaljs*)
This method is called every time a new HTML document is loaded in the viewer. Use it to load javascript libraries into the viewer. For example:

```
        def load_javascript(self, evaljs):
            js = get_resources('myjavascript.js')
            evaljs(js)
```

run_javascript(*evaljs*)
This method is called every time a document has finished loading. Use it in the same way as load_javascript().

customize_ui(*ui*)
This method is called once when the viewer is created. Use it to make any customizations you want to the viewer's user interface. For example, you can modify the toolbars via ui.tool_bar and ui.tool_bar2.

customize_context_menu(*menu*, *event*, *hit_test_result*)
This method is called every time the context (right-click) menu is shown. You can use it to customize the context menu. `event` is the context menu event and hit_test_result is the QWebHitTestResult for this event in the currently loaded document.

1.10.2 Proměnné prostředí

- `CALIBRE_CONFIG_DIRECTORY` – Nastaví adresář, kde jsou uložené konfigurační soubory.
- `CALIBRE_TEMP_DIR` – Nastaví dočasný adresář používaný Calibre.
- `CALIBRE_CACHE_DIRECTORY` – Nastaví adresář, který Calibre používá pro ukládání trvalých dat do mezipaměti mezi relacemi
- `CALIBRE_OVERRIDE_DATABASE_PATH` – Umožní vám zadat úplnou cestu k metadata.db. Pomocí této proměnné můžete mít metadata.db v jiném umístění než ve složce knihovny. Užitečné, pokud je vaše složka knihovny na síťové jednotce, která nepodporuje uzamykání souboru.
- `CALIBRE_DEVELOP_FROM` – Použito pro spuštění z vývojového prostředí Calibre. Podívejte se na *Nastavení vývojového prostředí calibre* (stránka 260).
- `CALIBRE_OVERRIDE_LANG` – Použito pro vynucení jazyka použitého rozhraním (kód jazyka ISO 639).
- `CALIBRE_TEST_TRANSLATION` - Použito pro testování souboru překladu .po (měla by to být cesta k souboru .po)
- `CALIBRE_NO_NATIVE_FILEDIALOGS` – Způsobí, že Calibre nepoužívá nativní dialogová okna souboru pro výběr souborů nebo adresářů. Nastavte na 1 pro povolení.
- `CALIBRE_NO_NATIVE_MENUBAR` – Způsobí, že Calibre nevytváří nativní (globální) nabídku v Ubuntu Unity a podobných desktopových prostředí Linux. Nabídka je místo toho umístěna do okna, jako obvykle.
- `CALIBRE_IGNORE_SYSTEM_THEME` – Způsobí, že Calibre bude ignorovat všechny systémové moduly stylu Qt a namísto toho použije svůj vestavěný modul stylu. Užitečné pro vyřešení pádů způsobených systémovým modulem Qt, který je nekompatibilní s verzí Qt dodanou s Calibre.
- `CALIBRE_SHOW_DEPRECATION_WARNINGS` – Způsobí, že Calibre vytiskne varování o neschválení do stdout. Užitečné pro vývojáře Calibre.
- `SYSFS_PATH` – Použijte, pokud je sysfs připojeno někde jinde než /sys.
- `http_proxy` – Použito v Linuxu pro zadání proxy HTTP.

Podívejte se na Jak nastavit proměnné prostředí ve Windows[90] nebo Jak nastavit proměnné prostředí v OS X[91].

[90] http://www.computerhope.com/issues/ch000549.htm
[91] http://www.dowdandassociates.com/blog/content/howto-set-an-environment-variable-in-mac-os-x-launchd-plist/

1.10.3 Vylepšení

Vylepšení jsou malé změny, které můžete zadat pro určení různých aspektů chování Calibre. Můžete je změnit v Předvolby->Rozšířené->Vylepšení. Výchozí hodnoty pro vylepšení jsou zopakovány níže

```
#!/usr/bin/env python2
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
__license__   = 'GPL v3'
__copyright__ = '2010, Kovid Goyal <kovid@kovidgoyal.net>'
__docformat__ = 'restructuredtext en'

'''
Contains various tweaks that affect calibre behavior. Only edit this file if
you know what you are doing. If you delete this file, it will be recreated from
defaults.
'''

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
# next - First available integer larger than the largest existing number
# first_free - First available integer larger than 0
# next_free - First available integer larger than the smallest existing number
# last_free - First available integer smaller than the largest existing number
#             Return largest existing + 1 if no free number is found
# const - Assign the number 1 always
# no_change - Do not change the series index
# a number - Assign that number always. The number is not in quotes. Note that
#            0.0 can be used here.
# Examples:
# series_index_auto_increment = 'next'
# series_index_auto_increment = 'next_free'
# series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, than that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Should the completion separator be append
# to the end of the completed text to
# automatically begin a new completion operation
# for authors.
# Can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort
```

```
# Possible values are:
#  invert: use "fn ln" -> "ln, fn"
#  copy  : copy author to author_sort without modification
#  comma : use 'copy' if there is a ',' in the name, otherwise use 'invert'
#  nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
# must be recomputed by right-clicking on an author in the left-hand tags pane,
# selecting 'manage authors', and pressing 'Recalculate all author sort values'.
# The author name suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled. The same is true for prefixes.
# The author name copy words are a set of words which if they occur in an
# author name cause the automatically generated author sort string to be
# identical to the author name. This means that the sort for a string like Acme
# Inc. will be Acme Inc. instead of Inc., Acme
author_sort_copy_method = 'comma'
author_name_suffixes = ('Jr', 'Sr', 'Inc', 'Ph.D', 'Phd',
                        'MD', 'M.D', 'I', 'II', 'III', 'IV',
                        'Junior', 'Senior')
author_name_prefixes = ('Mr', 'Mrs', 'Ms', 'Dr', 'Prof')
author_name_copywords = ('Corporation', 'Company', 'Co.', 'Agency', 'Council',
        'Committee', 'Inc.', 'Institute', 'Society', 'Club', 'Team')

#: Splitting multiple author names
# By default, calibre splits a string containing multiple author names on
# ampersands and the words "and" and "with". You can customize the splitting
# by changing the regular expression below. Strings are split on whatever the
# specified regular expression matches, in addition to ampersands.
# Default: r'(?i),?\s+(and|with)\s+'
authors_split_regex = r'(?i),?\s+(and|with)\s+'

#: Use author sort in Tag Browser
# Set which author field to display in the tags pane (the list of authors,
# series, publishers etc on the left hand side). The choices are author and
# author_sort. This tweak affects only what is displayed under the authors
# category in the tags pane and content server. Please note that if you set this
# to author_sort, it is very possible to see duplicate names in the list because
# although it is guaranteed that author names are unique, there is no such
# guarantee for author_sort values. Showing duplicates won't break anything, but
# it could lead to some confusion. When using 'author_sort', the tooltip will
# show the author's name.
# Examples:
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
categories_use_field_for_author_name = 'author'

#: Control partitioning of Tag Browser
# When partitioning the tags browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#  name: the printable name of the item
#  count: the number of books that references this item
```

```
#  avg_rating: the average rating of all the books referencing this item
#  sort: the sort value. For authors, this is the author_sort for that author
#  category: the category (e.g., authors, series) that the item is in.
# Note that the "r'" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.avg_rating:4.2f:
categories_collapsed_popularity_template = r'{first.count:d} - {last.count:d}'

#: Control order of categories in the tag browser
# Change the following dict to change the order that categories are displayed in
# the tag browser. Items are named using their lookup name, and will be sorted
# using the number supplied. The lookup name '*' stands for all names that
# otherwise do not appear. Two names with the same value will be sorted
# using the default order; the one used when the dict is empty.
# Example: tag_browser_category_order = {'series':1, 'tags':2, '*':3}
# resulting in the order series, tags, then everything else in default order.
tag_browser_category_order = {'*':1}


#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts
#  The argument is None if saved sort history is to be used
#  otherwise it is a list of column,order pairs. Column is the
#  lookup/search name, found using the tooltip for the column
#  Order is 0 for ascending, 1 for descending
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
#  A string controlling how the publication date is displayed in the GUI
#  d     the day as number without a leading zero (1 to 31)
#  dd    the day as number with a leading zero (01 to 31)
#  ddd   the abbreviated localized day name (e.g. 'Mon' to 'Sun').
#  dddd  the long localized day name (e.g. 'Monday' to 'Sunday').
#  M     the month as number without a leading zero (1-12)
#  MM    the month as number with a leading zero (01-12)
#  MMM   the abbreviated localized month name (e.g. 'Jan' to 'Dec').
#  MMMM  the long localized month name (e.g. 'January' to 'December').
#  yy    the year as two digit number (00-99)
#  yyyy  the year as four digit number
#  h     the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) '
#  hh    the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm) '
#  m     the minutes without a leading 0 (0 to 59) '
#  mm    the minutes with a leading 0 (00 to 59) '
#  s     the seconds without a leading 0 (0 to 59) '
#  ss    the seconds with a leading 0 (00 to 59) '
#  ap    use a 12-hour clock instead of a 24-hour clock, with "ap"
#        replaced by the localized string for am or pm '
#  AP    use a 12-hour clock instead of a 24-hour clock, with "AP"
#        replaced by the localized string for AM or PM '
#  iso   the date with time and timezone. Must be the only format present
#  For example, given the date of 9 Jan 2010, the following formats show
#  MMM yyyy ==> Jan 2010    yyyy ==> 2010       dd MMM yyyy ==> 09 Jan 2010
#  MM/yyyy ==> 01/2010      d/M/yy ==> 9/1/10   yy ==> 10
```

```
# publication default if not set: MMM yyyy
# timestamp default if not set: dd MMM yyyy
# last_modified_display_format if not set: dd MMM yyyy
gui_pubdate_display_format = 'MMM yyyy'
gui_timestamp_display_format = 'dd MMM yyyy'
gui_last_modified_display_format = 'dd MMM yyyy'

#: Control sorting of titles and series in the library display
# Control title and series sorting in the library view. If set to
# 'library_order', the title sort field will be used instead of the title.
# Unless you have manually edited the title sort field, leading articles such as
# The and A will be ignored. If set to 'strictly_alphabetic', the titles will be
# sorted as-is (sort by title instead of title sort). For example, with
# library_order, The Client will sort under 'C'. With strictly_alphabetic, the
# book will sort under 'T'.
# This flag affects Calibre's library display. It has no effect on devices. In
# addition, titles for books added before changing the flag will retain their
# order until the title is edited. Double-clicking on a title and hitting return
# without changing anything is sufficient to change the sort.
title_series_sorting = 'library_order'

#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# 'library_order', articles such as 'The' and 'An' will be moved to the end. If
# set to 'strictly_alphabetic', the series will be sent without change.
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
# available, the language of the book is used. You can change the list of
# articles for a given language or add a new language by editing
# per_language_title_sort_articles. To tell calibre to use a language other
# than the user interface language, set, default_language_for_title_sort. For
# example, to use German, set it to 'deu'. A value of None means the user
# interface language is used. The setting title_sort_articles is ignored
# (present only for legacy reasons).
per_language_title_sort_articles = {
        # English
        'eng'  : (r'A\s+', r'The\s+', r'An\s+'),

        # Esperanto
        'epo': (r'La\s+', r"L'", 'L\xb4'),

        # Spanish
        'spa'  : (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
                  r'Una\s+', r'Unos\s+', r'Unas\s+'),
```

```
        # French
        'fra'  : (r'Le\s+', r'La\s+', r"L'", u'L´', u'L’', r'Les\s+', r'Un\s+', r'Une\s+',
                  r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", u'D´', u'L’'),

        # Italian
        'ita': ('Lo\\s+', 'Il\\s+', "L'", 'L\xb4', 'La\\s+', 'Gli\\s+',
                'I\\s+', 'Le\\s+', 'Uno\\s+', 'Un\\s+', 'Una\\s+', "Un'",
                'Un\xb4', 'Dei\\s+', 'Degli\\s+', 'Delle\\s+', 'Del\\s+',
                'Della\\s+', 'Dello\\s+', "Dell'", 'Dell\xb4'),

        # Portuguese
        'por'  : (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
                  r'Uma\s+', r'Umas\s+', ),
        # Romanian
        'ron'  : (r'Un\s+', r'O\s+', r'Nişte\s+', ),
        # German
        'deu'  : (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
                  r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
                  r'Eines\s+'),
        # Dutch
        'nld'  : (r'De\s+', r'Het\s+', r'Een\s+', r"'n\s+", r"'s\s+", r'Ene\s+',
                  r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
                  r"'t\s+"),
        # Swedish
        'swe'  : (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+', ),
        # Turkish
        'tur'  : (r'Bir\s+', ),
        # Afrikaans
        'afr'  : (r"'n\s+", r'Die\s+', ),
        # Greek
        'ell'  : (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
                  r"'Enas\s+", r"'Mia\s+", r"'Ena\s+", r"'Enan\s+", ),
        # Hungarian
        'hun'  : (r'A\s+', 'Az\s+', 'Egy\s+',),
}
default_language_for_title_sort = None
title_sort_articles=r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored. If there are '\' characters in
# the path (such as in Windows paths), you must double them.
# Examples:
#     auto_connect_to_folder = 'C:\\Users\\someone\\Desktop\\testlib'
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''

#: Specify renaming rules for SONY collections
# Specify renaming rules for sony collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on Sonys are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field. For example, if
# the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
```

```
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection. For example, you
# could force the values in 'series', '#my_series_1', and '#my_series_2' to
# appear in collections named 'some_value (Series)', thereby merging all of the
# fields into one set of collections.
# There are two related tweaks. The first determines the category name to use
# for a metadata field.  The second is a template, used to determines how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
# {'field_lookup_name':'category_name_to_use', 'lookup_name':'name', ...}
# The second tweak, sony_collection_name_template, is a template. It uses the
# same template language as plugboards and save templates. This tweak controls
# how the value and category are combined together to make the collection name.
# The only two fields available are {category} and {value}. The {value} field is
# never empty. The {category} field can be empty. The default is to put the
# value first, then the category enclosed in parentheses, it isn't empty:
# '{value} {category:|(|)}'
# Examples: The first three examples assume that the second tweak
# has not been changed.
# 1: I want three series columns to be merged into one set of collections. The
# column lookup names are 'series', '#series_1' and '#series_2'. I want nothing
# in the parenthesis. The value to use in the tweak value would be:
#    sony_collection_renaming_rules={'series':'', '#series_1':'', '#series_2':''}
# 2: I want the word '(Series)' to appear on collections made from series, and
# the word '(Tag)' to appear on collections made from tags. Use:
#    sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
# 3: I want 'series' and '#myseries' to be merged, and for the collection name
# to have '(Series)' appended. The renaming rule is:
#    sony_collection_renaming_rules={'series':'Series', '#myseries':'Series'}
# 4: Same as example 2, but instead of having the category name in parentheses
# and appended to the value, I want it prepended and separated by a colon, such
# as in Series: Darkover. I must change the template used to format the category name
# The resulting two tweaks are:
#    sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#    sony_collection_name_template='{category:||: }{value}'
sony_collection_renaming_rules={}
sony_collection_name_template='{value}{category:| (|)}'

#: Specify how SONY collections are sorted
# Specify how sony collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series- based collection it is sorted by series order,
# otherwise it is sorted by title order.
# [(['pubdate', 'tags'],'#mydate'), (['series'],'series_index'), (['*'], 'title')]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [ list of fields ] , sort field ) ]
```

```
# Default: empty (no rules), so no collection attributes are named.
sony_collection_sorting_rules = []

#: Control how tags are applied when copying books to another library
# Set this to True to ensure that tags in 'Tags to add when adding
# a book' are added when copying books to another library
add_new_book_tags_when_importing_books = False

#: Set the maximum number of tags to show per book in the content server
max_content_server_tags_shown=5

#: Set custom metadata fields that the content server will or will not display.
# content_server_will_display is a list of custom fields to be displayed.
# content_server_wont_display is a list of custom fields not to be displayed.
# wont_display has priority over will_display.
# The special value '*' means all custom fields. The value [] means no entries.
# Defaults:
#    content_server_will_display = ['*']
#    content_server_wont_display = []
# Examples:
# To display only the custom fields #mytags and #genre:
#   content_server_will_display = ['#mytags', '#genre']
#   content_server_wont_display = []
# To display all fields except #mycomments:
#   content_server_will_display = ['*']
#   content_server_wont_display['#mycomments']
content_server_will_display = ['*']
content_server_wont_display = []

#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of doubleclicks on the books list.
# Choices: open_viewer, do_nothing,
# edit_cell, edit_metadata. Selecting anything other than open_viewer has the
# side effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls horizontal per column or
# per pixel. Default is per column.
```

```
doubleclick_on_library_view = 'open_viewer'
horizontal_scrolling_per_column = True

#: Language to use when sorting.
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# http://publib.boulder.ibm.com/infocenter/iseries/v5r3/topic/nls/rbagsicusortsequencetables.htm
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
locale_for_sorting =  ''

#: Number of columns for custom metadata in the edit metadata dialog
# Set whether to use one or two columns for custom metadata when editing
# metadata  one book at a time. If True, then the fields are laid out using two
# columns. If False, one column is used.
metadata_single_use_2_cols_for_custom_fields = True

#: Order of custom column(s) in edit metadata
# Controls the order that custom columns are listed in edit metadata single
# and bulk. The columns listed in the tweak are displayed first and in the
# order provided. Any columns not listed are dislayed after the listed ones,
# in alphabetical order. Do note that this tweak does not change the size of
# the edit widgets. Putting comments widgets in this list may result in some
# odd widget spacing when using two-column mode.
# Enter a comma-separated list of custom field lookup names, as in
# metadata_edit_custom_column_order = ['#genre', '#mytags', '#etc']
metadata_edit_custom_column_order = []

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like gmx/hotmail/gmail. Default is: 5 minutes
# Setting it to lower may cause the server's SPAM controls to kick in,
# making email sending fail. Changes will take effect only after a restart of
# calibre.
public_smtp_relay_delay = 301

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = "main"

#: What interfaces should the content server listen on
# By default, the calibre content server listens on '0.0.0.0' which means that it
# accepts IPv4 connections on all interfaces. You can change this to, for
# example, '127.0.0.1' to only listen for connections from the local machine, or
```

```
# to '::' to listen to all incoming IPv6 and IPv4 connections (this may not
# work on all operating systems)
server_listen_on = '0.0.0.0'

#: Unified toolbar on OS X
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for OS X applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the View button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of book details in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: Compile General Program Mode templates to Python
# Compiled general program mode templates are significantly faster than
# interpreted templates. Setting this tweak to True causes calibre to compile
# (in most cases) general program mode templates. Setting it to False causes
# calibre to use the old behavior -- interpreting the templates. Set the tweak
# to False if some compiled templates produce incorrect values.
# Default:    compile_gpm_templates = True
# No compile: compile_gpm_templates = False
compile_gpm_templates = True

#: What format to default to when using the Tweak feature
# The Tweak feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
# Set to a value of 'remember' to use whichever format you chose last time you
# used the Tweak feature.
# Examples:
#   default_tweak_format = None       (Use output format)
#   default_tweak_format = 'EPUB'
#   default_tweak_format = 'remember'
default_tweak_format = None

#: Do not preselect a completion when editing authors/tags/series/etc.
```

```
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
# completions you will now have to press Tab to select one before pressing
# Enter. Which technique you prefer will depend on the state of metadata in
# your library and your personal editing style.
preselect_first_completion = False

#: Completion mode when editing authors/tags/series/etc.
# By default, when completing items, calibre will show you all the candidates
# that start with the text you have already typed. You can instead have it show
# all candidates that contain the text you have already typed. To do this, set
# completion_mode to 'contains'. For example, if you type asi it will match both
# Asimov and Quasimodo, whereas the default behavior would match only Asimov.
completion_mode = 'prefix'

#: Recognize numbers inside text when sorting
# This means that when sorting on text fields like title the text "Book 2"
# will sort before the text "Book 100". If you want this behavior, set
# numeric_collation = True note that doing so will cause problems with text
# that starts with numbers and is a little slower.
numeric_collation = False

#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to Library and Quick Switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10

#: Highlight the virtual library name when using a Virtual Library
# The virtual library name next to the Virtual Library button is highlighted in
# yellow when using a Virtual Library. You can choose the color used for the
# highlight with this tweak. Set it to 'transparent' to disable highlighting.
highlight_virtual_library = 'yellow'

#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# For example, if you only want to convert to EPUB and AZW3, change this to
# restrict_output_formats = ['EPUB', 'AZW3']. The default value of None causes
# all available output formats to be present.
restrict_output_formats = None

#: Set the thumbnail image quality used by the content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75
```

1.10.4 Přepsání ikon, šablon atd.

Poznámka: Calibre má přímou podporu pro motivy ikon, je pro něj dostupných několik motivů ikon, které můžete použít přejitím na *Předvolby->Vzhled->Změnit motiv ikon*. Motivy ikon používají stejný mechanismus, jak je popsáno níže pro přepsání statických zdrojů.

Calibre vám umožňuje přepsat statické zdroje, jako jsou ikony, javascript a šablony, pro přebal s metadaty, katalogy atd. upravenými verzemi, které se vám líbí. Všechny statické zdroje jsou uložené v podsložce resources instalačního umístění Calibre. Ve Windows je to obvykle `C:\Program Files\Calibre2resources`. V OS X `/Applications/calibre.app/Contents/Resources/resources/`. V Linuxu, pokud používáte binární instalační balíček z webové stránky Calibre, to bude `/opt/calibre/resources`. Tyto cesty se můžou měnit v závislosti na tom, kam si zvolíte nainstalovat Calibre.

Soubory v této složce zdrojů byste neměli měnit, protože vaše změny budou přepsaány při příští aktualizaci Calibre. Namísto toho přejděte do *Předvolby->Rozšířené->Různé* a klikněte na *Otevřít konfigurační adresář Calibre*. V tomto konfiguračním adresáři vytvořte podsložku nazvanou resources a umístěte do ní soubory, které chcete přepsat. Umístěte soubory do příslušných podsložek, například obrázky umístěte do `resources/images` atd. Calibre automaticky použije při příštím spuštění váš vlastní soubor přednostně před vestavěným.

Například pokud chcete změnit ikonu pro akci *Odebrat knihy*, nejdříve byste se podívali do vestavěné složky resources a viděli byste, že odpovídající soubor je `resources/images/trash.png`. Pokud máte alternativní ikonu ve formátu PNG nazvanou `mytrash.png`, uložili byste ji do konfiguračního adresáře jako `resources/images/trash.png`. Všechny ikony použité uživatelským rozhraním Calibre jsou v `resources/images` a jeho podsložkách.

1.10.5 Vytvoření vlastního motivu ikon pro Calibre

Pokud jste vytvořili krásnou sadu ikon a chcete ji sdílet s ostatními uživateli Calibre prostřednictvím vestavěného podpory motivů ikon Calibre, můžete snadno zabalit vaše ikony do motivu. K tomu přejděte na *Předvolby->Různé->Vytvořit motiv ikon*, vyberte složku, kam jste vložili své ikony (obvykle složka `resources/images` v konfiguračním adresáři Calibre, jak je popsáno výše). Pak vyplňte metadata motivu a klikněte na OK. Vznikne tak soubor zip obsahující ikony motivu. Ten můžete nahrát na fórum Calibre na Mobileread[92] a já pak zpřístupním váš motiv prostřednictvím vestavěného systému motivu ikon Calibre.

1.10.6 Přizpůsobení Calibre pomocí modulů

Calibre má velice modulární provedení. Téměř všechny funkce jsou ve formě modulů. Moduly jsou použity pro převod, stahování zpráv (ačkoliv ty jsou nazvány předpisy), různé komponenty uživatelského rozhraní, připojení k různým zařízením, zpracování souborů při jejich přidávání do Calibre a tak dále. Kompletní seznam všech vestavěných modulů můžete získat v Calibre v *Předvolby->Moduly*.

Pro přizpůsobení a rozšíření chování Calibre můžete napsat své vlastní moduly. Architektura modulu v Calibre je velice jednoduchá, podívejte se na kurz *Psaní vlastních modulů pro rozšíření funkcí Calibre* (stránka 170).

1.11 Rozhraní příkazového řádku

Poznámka: Na OS X jsou nástroje příkazového řádku uvnitř sady calibre, například pokud jste nainstalovali Calibre do `/Applications`, nástroje příkazového řádku jsou v `/Applications/calibre.app/Contents/console.app/Contents/MacO`

[92] http://www.mobileread.com/forums/forumdisplay.php?f=166

1.11.1 Dokumentované příkazy

calibre

```
calibre [opts] [path_to_ebook]
```

Spustit hlavní grafické uživatelské rozhraní Calibre a volitelně přidat e-knihu z path_to_ebook do databáze.

Kdykoliv předáte parametry **calibre**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--detach
Odpojit od řídícího terminálu, pokud nějaký je (pouze pro Linux)

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--ignore-plugins
Ignorovat vlastní moduly. Užitečné, pokud jste nainstalovali modul, který brání Calibre ve spuštění.

--no-update-check
Nekontrolovat aktualizace

--shutdown-running-calibre, -s
Způsobí vypnutí spuštěných instancí Calibre. Pamatujte, že pokud máte spuštěné úlohy, budou bez upozornění přerušeny, takže to používejte opatrně.

--start-in-tray
Spustit minimalizované do oznamovací oblasti hlavního panelu.

--verbose, -v
Ignorováno, nepoužívejte. Přítomno pouze z historických důvodů

--version
zobrazit číslo verze programu a ukončit

--with-library
Použít knihovnu umístěnou v zadané cestě.

calibre-customize

```
calibre-customize options
```

Přizpůsobit Calibre načtením externích modulů.

Kdykoliv předáte parametry **calibre-customize**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--add-plugin, -a
Přidat modul zadáním cesty k souboru zip, který ho obsahuje.

--build-plugin, -b
Pro vývojáře modulů: Cesta k adresáři, kde vyvíjíte modul. Tento příkaz automaticky zabalí modul a aktualizuje ho v Calibre.

--customize-plugin
Přizpůsobit modul. Zadejte název modulu a řetězec s úpravami oddělenými čárkou.

--disable-plugin
Zakázat uvedený modul

--enable-plugin
Povolit uvedený modul

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--list-plugins, -l
Vypsat všechny nainstalované moduly

--remove-plugin, -r
Odebrat vlastní modul podle jména. Nemá žádný vliv na vestavěné moduly.

--version
zobrazit číslo verze programu a ukončit

calibre-debug

```
calibre-debug [options]
```

Různá rozhraní příkazového řádku užitečná pro ladění Calibre. Bez voleb spouští tento příkaz vložený překladač Pythonu. Můžete také spustit hlavní grafické rozhraní Calibre, prohlížeč Calibre a editor Calibre v režimu ladění.

Také obsahuje rozhraní pro různé kusy Calibre, které neobsahují vyhrazené nástroje příkazového řádku, jako jsou částečné vkládání písma, nástroj rozdílů e-knih a tak dále.

Můžete také použít **calibre-debug** ke spuštění samostatných skriptů. K tomu ho použijte takto:

calibre-debug myscript.py -- --option1 --option2 file1 file2 ...

Všechno za -- je předáno skriptu.

Kdykoliv předáte parametry **calibre-debug**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--add-simple-plugin
Přidat jednoduchý doplněk (např. doplněk obsahující pouze soubor .py) zadáním cesty k souboru py obsahujícímu kód doplňku.

--command, -c
Spustit kód Pythonu.

--debug-device-driver, -d
Ladit rozpoznávání zařízení

--default-programs
(Od)registrovat Calibre ve výchozích programech Windows. --default-programs=(register|unregister)

--diff
Spustit rozdílový nástroj Calibre. Například: calibre-debug *--diff* (stránka 230) soubor1 soubor2

--edit-book, -t
Spustit nástroj calibre Upravit knihu v režimu ladění.

--exec-file, -e
Spustit kód Pythonu v souboru.

--explode-book, -x
Rozložit knihu (exportuje knihu jako kolekci souborů HTML a metadat, které můžete upravovat pomocí standardních nástrojů pro úpravu HTML, a poté znovu sestaví soubor z upraveného HTML. Nedělá žádné dodatečné změny v HTML na rozdíl od úplného převodu pomocí Calibre).

--gui, -g
Spustit grafické rozhraní s povoleným laděním. Výstup ladění je vytištěn do stdout a stderr.

--gui-debug
Spustit grafické rozhraní s konzolí ladění, protokolování do zadané cesty. Pouze pro vnitřní použití, pro spuštění grafického rozhraní v režimu ladění použijte volbu -g

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--inspect-mobi, -m
Zkontrolovat soubory MOBI v zadaných cestách

--new-server
Run the new calibre content server. Any options specified after a `--` will be passed to the server.

--paths
Výstup cest potřebných pro nastavení prostředí Calibre

--py-console, -p
Spustit konzoli Pythonu

--reinitialize-db
Znovu inicializovat SQLite databázi Calibre v zadané cestě. Užitečné pro obnovení po poškození databáze.

--run-plugin, -r
Spustit modul poskytující rozhraní příkazového řádku. Například: calibre-debug -r "Přidat knihy" `--` soubor1 `--option1` Všechno za `--` bude předáno modulu jako parametry.

--shutdown-running-calibre, -s
Způsobí vypnutí spuštěných instancí Calibre. Pamatujte, že pokud máte spuštěné úlohy, budou bez upozornění přerušeny, takže to používejte opatrně.

--subset-font, -f
Částečně vložit zadané písmo. Použijte `--` po této volbě pro předání volby programu pro částečné vkládání písem.

--test-build
Testovat binární moduly v sestavení

--version
zobrazit číslo verze programu a ukončit

--viewer, -w
Spustit prohlížeč e-knih v režimu ladění.

calibre-server

```
calibre-server [options]
```

Spustit server s obsahem Calibre. Server s obsahem Calibre vystaví vaši knihovnu Calibre na internet. Výchozí rozhraní vám umožňuje procházez knihovnu Calibre podle kategorií. Můžete také přistupovat k rozhraní optimalizovanému pro mobilní prohlížeče na /mobile a k rozhraní na základě OPDS pro použití se čtecími aplikacemi na /opds.

Rozhraní OPDS je ohlášeno přes BonJour automaticky.

Kdykoliv předáte parametry **calibre-server**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--auto-reload
Automaticky znovu načíst server při změně zdrojového kódu. Nemusí fungovat ve všech prostředích.

--daemonize
Spustit proces na pozadí jako démon. Nemá žádný vliv na okna.

--develop
Vývojářský režim. Server se automaticky restartuje při změnách souborů a dodá soubory kódu (html, css, js) ze systému souborů namísto ze systému prostředků Calibre.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--max-cover
Maximální velikost pro zobrazené obálky. Výchozí je '600x800'.

--max-opds-items
Nejvyšší počet shod vrácených na dotaz OPDS. Toto ovlivní integrace Stanza, WordPlayer atd.

--max-opds-ungrouped-items
Seskupit položky v kategoriích, jako jsou autoři nebo štítky, podle prvního písmena, když je více než tento počet položek. Výchozí: 100. Nastavte na velké číslo pro zakázání seskupování.

--password
Nastavit heslo pro omezení přístupu. Ve výchozím nastavení je přístup bez omezení.

--pidfile
Zapsat kód PID procesu do zadaného souboru

--port, -p
Port, na kterém naslouchat. Výchozí je 8080

--restriction
Určuje virtuální knihovnou použitou pro toto vyvolání. Tato volba přepíše jakákoliv nastavení pro knihovnu zadaná v grafickém rozhraní. Pokud hodnota není virtuální knihovnou, ale je uloženým hledáním, je pro kompatibilitu použito toto uložené hledání. Také pamatujte, že pokud nezadáte omezení, bude použita hodnota zadaná v grafickém rozhraní (pokud nějaká je).

--thread-pool
Maximální počet použitých pracovních vláken. Výchozí je 30

--timeout, -t
Časový limit serveru v sekundách. Výchozí je 120

--url-prefix
Předpona pro předřazení ke všem URL. Užitečné pro reverzní proxy na tento server z Apache, nginx atd.

--username
Uživatelské jméno pro přístup. Ve výchozím nastavení je to: 'calibre'

--version
zobrazit číslo verze programu a ukončit

--with-library
Cesta ke složce knihovny sloužící pro server s obsahem

calibre-smtp

```
calibre-smtp [options] [from to text]
```

Odeslat e-mail pomocí protokolu SMTP. **calibre-smtp** má dva režimy. V režimu psaní zadáte from to a text, a ty jsou použity k sestavení a odeslání e-mailové zprávy. V režimu filtrování čte **calibre-smtp** kompletní e-mailovou zprávu ze vstupu STDIN a odešle ji.

text je tělo e-mailové zprávy. Pokud text není zadán, kompletní e-mailová zpráva je čtena ze vstupu STDIN. from je e-mailová adresa odesílatele a to je e-mailová adresa příjemce. Když je kompletní e-mail čten ze vstupu STDIN, from a to jsou použity pouze při připojení vstupu STDIN, hlavičky zpráv nejsou měněny.

Kdykoliv předáte parametry **calibre-smtp**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--fork, -f
Rozvětvit a doručit zprávu na pozadí. Pokud použijete tuto volbu, měli byste také použít *--outbox* (stránka 233) pro vyřešení selhání doručení.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--localhost, -l
Název hostitele localhost. Použito při připojování k serveru SMTP.

--outbox, -o
Cesta ke složce maildir pro e-mailové zprávy, které se nepodařilo odeslat.

--timeout, -t
Časový limit pro připojení

--verbose, -v
Více podrobností

--version
zobrazit číslo verze programu a ukončit

COMPOSE MAIL Volby pro napsání e-mailu. Ignorovat, pokud text není zadán

--attachment, -a
Soubor připojený k e-mailu

--subject, -s
Předmět e-mailu

SMTP RELAY Volby pro použití serveru přenosu SMTP pro odesílání pošty. Calibre se pokusí odeslat e-mail přímo, pokud není zadáno –relay.

--encryption-method, -e
Metoda šifrování použitá při připojování k přenosu. Možnosti jsou TLS, SSL a Žádné. Výchozí je TLS. VAROVÁNÍ: Výběr volby Žádné je vysoce nezabezpečený.

--password, -p
Heslo pro přenos

--port
Port pro připojení k serveru přenosu. Výchozí je použít 465, pokud je metoda šifrování SSL, jinak 25.

--relay, -r
Server přenosu SMTP použitý pro odeslání e-mailu.

--username, -u
Uživatelské jméno pro přenos

calibredb

```
calibredb command [options] [arguments]
```

calibredb is the command line interface to the calibre database. It has several sub-commands, documented below:

- *Globální volby* (stránka 234)
- *list* (stránka 235)
- *add* (stránka 235)
- *remove* (stránka 236)
- *add_format* (stránka 237)
- *remove_format* (stránka 237)
- *show_metadata* (stránka 237)
- *set_metadata* (stránka 238)
- *export* (stránka 238)
- *catalog* (stránka 239)
- *saved_searches* (stránka 240)
- *add_custom_column* (stránka 240)
- *custom_columns* (stránka 241)
- *remove_custom_column* (stránka 241)
- *set_custom* (stránka 241)
- *restore_database* (stránka 242)
- *check_library* (stránka 242)
- *list_categories* (stránka 242)
- *backup_metadata* (stránka 243)
- *clone* (stránka 243)
- *embed_metadata* (stránka 244)
- *search* (stránka 244)

Globální volby

--dont-notify-gui
Neupozorňovat spuštěné grafické rozhraní Calibre (pokud je nějaké), že se databáze změnila. Používejte opatrně, protože to může vést k poškození databáze!

--library-path
Cesta ke knihovně Calibre. Výchozí je použít cesta uloženou v nastavení.

list

```
calibredb list [options]
```

Vypsat knihy dostupné v databázi Calibre.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--ascending
Seřadit výsledky ve vzestupném pořadí

--fields, -f
Pole k zobrazení při výpisu knih v databázi. Měl by to být čárkami oddělený seznam polí. Dostupná pole: author_sort, authors, comments, cover, formats, id, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, timestamp, title, uuid Výchozí: title,authors. Speciální pole "all" může být použito pro výběr všech polí.

--for-machine
Generovat výstup ve formátu JSON, který je vhodnější pro strojní analýzu. Způsobí, že šířka řádku a volby oddělovače budou ignorovány.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--limit
Maximální počet výsledků k zobrazení. Výchozí: vše

--line-width, -w
Maximální šířka jednoho řádku ve výstupu. Výchozí nastavení pro určení velikosti obrazovky.

--prefix
Předpona pro všechny cesty k souborům. Výchozí je absolutní cesta do složky knihovny.

--search, -s
Filtrovat výsledky podle vyhledávacího dotazu. Pro formát vyhledávacího dotazu se podívejte do dokumentace související s hledáním v uživatelské příručce. Výchozí je neprovádět žádné filtrování.

--separator
Řetězec použitý pro oddělení polí. Výchozí je mezera.

--sort-by
Pole, podle kterého řadit výsledky. Dostupná pole: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, timestamp, title, uuid Výchozí: id

--version
zobrazit číslo verze programu a ukončit

add

```
calibredb add [options] file1 file2 file3...
```

Přidat zadané soubory jako knihy do databáze. Můžete také zadat adresáře, podívejte se na volby související s následujícím adresářem.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

--authors, -a
Nastavit autory přidaných knih

--cover, -c
Nastavit použitou obálku pro přidanou knihu

--duplicates, -d
Přidat knihy do databáze, i když již existují. Porovnání je prováděno na základě názvů knih.

--empty, -e
Přidat prázdnou knihu (kniha bez formátů)

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--identifier, -I
Nastavit identifikátory pro tuto knihu, např. -I asin:XXX -I isbn:YYY

--isbn, -i
Nastavit ISBN přidaných knih

--languages, -l
Čárkami oddělený seznam jazyků (nejlepší je použít kódy jazyka ISO639, i když mohou být rozpoznány také některé názvy jazyků)

--one-book-per-directory, -1
Předpokládat, že každý adresář má pouze jednu logickou knihu, a že všechny soubory v něm jsou různé formáty e-knih této knihy

--recurse, -r
Zpracovat adresáře rekurzivně

--series, -s
Nastavit série přidaných knih

--series-index, -S
Nastavit číslo série přidaných knih

--tags, -T
Nastavit štítky přidaných knih

--title, -t
Nastavit název přidaných knih

--version
zobrazit číslo verze programu a ukončit

remove

```
calibredb remove ids
```

Odebrat knihy identifikované identifikátory z databáze. Identifikátory by měly být čárkami oddělený seznam identifikačních čísel (identifikační čísla můžete získat pomocí příkazu search). Například 23,34,57-85 (když zadáváte rozsah, poslední číslo rozsahu není zahrnuto).

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

add_format

```
calibredb add_format [options] id ebook_file
```

Přidat e-knihu v ebook_file do dostupných formátů pro logickou knihu identifikovanou identifikátorem. Identifikátor můžete získat pomocí příkazu search. Pokud formát již existuje, je nahrazen, pokud není zadána volba nenahrazovat.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--dont-replace
Nenahrazovat formát, pokud již existuje

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

remove_format

```
calibredb remove_format [options] id fmt
```

Odebrat formát fmt z logické knihy identifikované identifikátorem. Identifikátor můžete získat pomocí příkazu search. fmt by měla být přípona souboru, jako je LRF, TXT nebo EPUB. Pokud logická kniha nemá dostupné fmt, nedělat nic.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

show_metadata

```
calibredb show_metadata [options] id
```

Zobrazit metadata uložená v databázi Calibre pro knihu identifikovanou identifikátorem. Identifikátor je identifikační číslo z příkazu search.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--as-opf
Vypsat metadata v podobě OPF (XML)

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

set_metadata

```
calibredb set_metadata [options] id [/path/to/metadata.opf]
```

Nastavit metadata uložená v databázi Calibre pro knihu identifikovanou identifikátorem ze souboru OPF metadata.opf. Identifikátor je identifikační číslo z příkazu search. Můžete získat rychlé chování pro formát OPF pomocí přepínače –as-opf příkazu show_metadata. Můžete také nastavit metadata jednotlivých polí pomocí volby –field. Pokud použijete volbu –field, není potřeba zadávat soubor OPF.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--field, -f
Pole k nastavení. Formát je field_name:value, například : *--field* (stránka 238) tags:tag1,tag2. Použijte *--list-fields* (stránka 238) pro získání seznamu všech názvů polí. Tuto volbu můžete zadat vícekrát pro nastavení více polí. Poznámka: Pro jazyky musíte použít kódy jazyka ISO639 (např. cs pro češtinu, en pro angličtinu, fr pro francouzštinu atd.). Pro identifikátory je syntaxe *--field* (stránka 238) identifiers:isbn:XXXX,doi:YYYYY. Pro pole logických hodnot (ano/ne) použijte pravda a nepravda nebo ano a ne.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--list-fields, -l
Vypsat názvy polí metadat, která mohou být použita s volbou *--field* (stránka 238)

--version
zobrazit číslo verze programu a ukončit

export

```
calibredb export [options] ids
```

Exportovat knihy zadané identifikátory (čárkami oddělený seznam) do systému souborů. Operace exportu ukládá všechny formáty knihy, její obálku a metadata (do souboru opf). Identifikační čísla můžete získat z příkazu search.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--all
Exportovat všechny knihy v databázi, ignorovat seznam identifikátorů.

--dont-asciiize
Obyčejně Calibre převede všechny neanglické znaky na anglické ekvivalenty pro názvy souborů. VAROVÁNÍ: Pokud toto vypnete, můžete zažít chyby při ukládání v závislosti na tom, jak dobře systém souborů, do kterého ukládáte, podporuje unicode. Zadání tohoto přepínače vypne toto chování.

--dont-save-cover
Obyčejně Calibre uloží obálku do samostatného souboru společně se skutečnými soubory e-knih. Zadání tohoto přepínače vypne toto chování.

--dont-update-metadata
Obyčejně Calibre aktualizuje metadata v uložených souborech z toho, co je v knihovně Calibre. Zpomalí ukládání na disk. Zadání tohoto přepínače vypne toto chování.

--dont-write-opf
Obyčejně Calibre zapíše metadata do samostatného souboru OPF společně se skutečnými soubory e-knih. Zadání tohoto přepínače vypne toto chování.

--formats
Čárkami oddělený seznam formátů pro uložení pro každou knihu. Ve výchozím nastavení jsou ukládány všechny dostupné formáty.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--replace-whitespace
Nahradit prázdné znaky podtržítky.

--single-dir
Exportovat všechny knihy do jednoho adresáře

--template
Šablona pro ovládání názvu souboru a adresářové struktury ukládaných souborů. Výchozí je "{author_sort}/{title}/{title} - {authors}", což uloží knihy do podadresáře podle autora s názvy souborů obsahujícími název a autora. Dostupné ovládací prvky jsou: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt
Formát, ve kterém zobrazovat data. %d - den, %b - měsíc, %m - číslo měsíce, %Y - rok. Výchozí je: %b, %Y

--to-dir
Exportovat knihy do zadaného adresáře. Výchozí je .

--to-lowercase
Převést cesty na malá písmena.

--version
zobrazit číslo verze programu a ukončit

catalog

```
calibredb catalog /path/to/destination.(CSV|EPUB|MOBI|XML ...) [options]
```

Exportovat katalog ve formátu zadaném rozšířením path/to/destination. Volby řídí, jak jsou položky zobrazeny v generovaném výstupním katalogu.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--fields
Pole pro výstup při zařazení knih do databáze. Měl by být čárkami oddělený seznam polí. Dostupná pole: all, title, title_sort, author_sort, authors, comments, cover, formats, id, isbn, library_name, ondevice, pubdate, publisher, rating, series_index, series, size, tags, timestamp, uuid, languages, identifiers, navíc uživatelem vytvořená vlastní pole. Příklad: `--fields=title,authors,tags` Výchozí: 'all' Použije se na: výstupní formáty CSV, XML

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--ids, -i
Čárkami oddělený seznam identifikátorů databáze pro katalog. Pokud je deklarováno, *--search* (stránka 239) je ignorováno. Výchozí: vše

--search, -s
Filtrovat výsledky podle vyhledávacího dotazu. Pro formát vyhledávacího dotazu se podívejte do dokumentace související s hledáním v uživatelské příručce. Výchozí: žádné filtrování

--sort-by
Výstupní pole, podle kterých řadit. Dostupná pole: author_sort, id, rating, size, timestamp, title_sort. Výchozí: 'id' Použije se na: výstupní formáty CSV, XML

--verbose, -v
Zobrazit podrobné výstupní informace. Užitečné pro ladění

--version
zobrazit číslo verze programu a ukončit

saved_searches

```
calibredb saved_searches [options] list
```

calibredb **saved_searches** add name search calibredb **saved_searches** remove name

Spravovat uložená hledání v databázi. Pokud se pokusíte přidat dotaz s názvem, který již existuje, bude nahrazen.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

add_custom_column

```
calibredb add_custom_column [options] label name datatype
```

Vytvořit vlastní sloupec. label je název sloupce vhodný pro stroje. Neměl by obsahovat mezery nebo dvojtečky. name je název sloupce vhodný pro lidi. datatype je jeden z: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--display
Slovník voleb pro přizpůsobení, jak budou data v tomto sloupci interpretována. Toto je řetězec JSON. Pro sloupce výčtu použijte `--display`="{"enum_values":["val1", "val2"]}" Je mnoho voleb, které mohou přijít do proměnné display. Volby podle typu sloupce jsou: složené: composite_template, composite_sort, make_category,contains_html, use_decorations datum a čas: date_format výčet: enum_values, enum_colors, use_decorations celá část, plovoucí: number_format text: is_names, use_decorations Nejlepší způsob jak najít zákonité kombinace, je vytvořit v grafickém rozhraní vlastní sloupec vhodného typu, pak vyhledat v záložním OPF knihu (ujistěte se, že byl vytvořen nový OPF od přidání sloupce). Uvidíte JSON pro "display" pro nový sloupec v OPF.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--is-multiple
Tento sloupec ukládá štítky jako data (tj. více čárkami oddělených hodnot). Použije se, pouze pokud je datový typ text.

--version
zobrazit číslo verze programu a ukončit

custom_columns

```
calibredb custom_columns [options]
```

Vypsat dostupné vlastní sloupce. Zobrazí popisky sloupců a identifikátory.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--details, -d
Zobrazit podrobnosti pro každý sloupec.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

remove_custom_column

```
calibredb remove_custom_column [options] label
```

Odebrat vlastní sloupec identifikovaný popiskem. Dostupné sloupce můžete zobrazit příkazem custom_columns.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--force, -f
Nepožadovat potvrzení

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

set_custom

```
calibredb set_custom [options] column id value
```

Nastavit hodnotu vlastního sloupce pro knihu identifikovanou identifikátorem. Seznam identifikátorů můžete získat pomocí příkazu search. Seznam názvů vlastních sloupců můžete získat pomocí příkazu custom_columns.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--append, -a
Pokud sloupec ukládá více hodnot, přidat zadané hodnoty k existujícím, namísto jejich nahrazení.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

restore_database

```
calibredb restore_database [options]
```

Obnovit tuto databázi z metadat uložených v souborech OPF v každém adresáři knihovny Calibre. To je užitečné, pokud byl váš soubor meatadata.db poškozen.

VAROVÁNÍ: Tento příkaz znovu úplně vygeneruje vaši databázi. Ztratíte všechna uložená hledání, uživatelské katalogy, zásuvné panely, uložená nastavení převodu pro knihu a vlastní předpisy. Obnovená metadata budou pouze tak přesná, jako to, co je nalezeno v souborech OPF.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--really-do-it, -r
Opravdu provést obnovení. Příkaz nebude spuštěn, pokud není zadána tato volba.

--version
zobrazit číslo verze programu a ukončit

check_library

```
calibredb check_library [options]
```

Provést některé kontroly souborového systému představujícího knihovnu. Hlášení jsou invalid_titles, extra_titles, invalid_authors, extra_authors, missing_formats, extra_formats, extra_files, missing_covers, extra_covers, failed_folders

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

--csv, -c
Výstup v CSV

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--ignore_extensions, -e
Čárkami oddělený seznam ignorovaných přípon. Výchozí: vše

--ignore_names, -n
Čárkami oddělený seznam ignorovaných názvů. Výchozí: vše

--report, -r
Čárkami oddělený seznam hlášení. Výchozí: vše

--version
zobrazit číslo verze programu a ukončit

list_categories

```
calibredb list_categories [options]
```

Vytvořit hlášení informací o kategoriích v databázi. Informace je ekvivalentem toho, co je zobrazeno v podokně štítků.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

--categories, -r
Čárkami oddělený seznam kategorií názvů vyhledávání. Výchozí: vše

--csv, -c
Výstup v CSV

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--item_count, -i
Výstupem je pouze počet položek v kategorii namísto počtů na položku v kategorii.

--quote, -q
Znak umístění kolem hodnoty kategorie v režimu CSV. Výchozí je uvozovky (").

--separator, -s
Řetězec použitý k oddělení polí v režimu CSV. Výchozí je čárka.

--version
zobrazit číslo verze programu a ukončit

--width, -w
Maximální šířka jednoho řádku ve výstupu. Výchozí nastavení pro určení velikosti obrazovky.

backup_metadata

```
calibredb backup_metadata [options]
```

Zálohovat metadata uložená v databázi do jednotlivých souborů OPF v každém adresáři knih. Toto se normálně děje automaticky, ale můžete tento příkaz spustit s volbou –all, abyste vynutili opětovné vygenerování souborů OPF.

Pamatujte, že normálně není vůbec potřeba toto dělat, protože soubory OPF jsou zálohovány automaticky pokaždé, když se změní metadata.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--all
Normálně tento příkaz pracuje pouze nad knihami, které mají zastaralé soubory OPF. Pomocí této volby pracuje nad všemi knihami.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

clone

```
calibredb clone path/to/new/library
```

Vytvořit klon aktuální knihovny. Toto vytvoří novou, prázdnou knihovnu, která má stejné vlastní sloupce, virtuální knihovny a jiná nastavení jako aktuální knihovna.

Klonovaná knihovna nebude obsahovat žádné knihy. Pokud chcete vytvořit úplný duplikát včetně všech knih, pak jednoduše použijte nástroje vašeho systému souborů ke zkopírování složky knihovny.

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--version
zobrazit číslo verze programu a ukončit

embed_metadata

```
calibredb embed_metadata [options] book_id
```

Aktualizovat metadata ve skutečných souborech knih uložená v knihovně Calibre z metadat v databázi Calibre. Normálně jsou metadata aktualizována pouze při exportování souborů z Calibre, tento příkaz je užitečný, pokud chcete, aby byla metadata aktualizována na místě. Pamatujte, že různé formáty souborů podporují různá množství metadat. K aktualizaci metadat ve všech knihách můžete pro book_id použít speciální hodnotu 'all'. Můžete také zadat mnoho identifikátorů knih oddělených mezerami a rozsahů identifikátorů oddělených spojovníky. Například: calibredb **embed_metadata** 1 2 10-15 23

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--only-formats, -f
Aktualizovat metadata pouze v souborech zadaného formátu. Zadejte vícekrát pro více formátů. Ve výchozím nastavení jsou aktualizovány všechny formáty.

--version
zobrazit číslo verze programu a ukončit

search

```
calibredb search [options] search expression
```

Vyhledat v knihovně zadaný hledaný výraz vracející čárkami oddělený seznam identifikátorů knih odpovídající hledanému výrazu.Výstupní formát je užitečný pro zásobení jiných příkazů, které přijímají seznam identifikátorů jako vstup.

Hledaný výraz může být cokoliv z výkonného jazyka vyhledávacích dotazů Calibre, například: author:asimov title:robot

Kdykoliv předáte parametry calibredb, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--limit, -l
Maximální počet výsledků k navrácení. Výchozí je všechny výsledky.

--version
zobrazit číslo verze programu a ukončit

ebook-convert

```
ebook-convert vstupni_soubor vystupni_soubor [volby]
```

Převeďte e-knihu z jednoho formátu do jiného.

vstupni_soubor je vstup a vystupni_soubor je výstup. Oba musí být zadány jako první dva parametry příkazu.

Výstupní formát e-knihy je odhadnut z příponou vystupni_soubor. vystupni_soubor může být také speciálního formátu .EXT, kde EXT je přípona výstupního souboru. V tomto případě je název výstupního souboru odvozen od názvu vstupního souboru. Pamatujte, že názvy souborů nesmí začínat pomlčkou. A konečně, pokud vystupni_soubor nemá žádnou příponu, pak je s ním zacházeno jako s adresářem a do tohoto adresáře je zapsána "otevřená e-kniha" (OEB) sestávající ze souborů HTML. Tyto soubory jsou soubory, které by obyčejně byly předány výstupnímu modulu.

Po zadání vstupního a výstupního souboru můžete přizpůsobit převod zadáním různých voleb. Dostupné volby závisí na typech vstupních a výstupních souborů. Pro získání nápovědy k nim zadejte vstupní a výstupní soubor a použijte volbu -h.

Úplnou dokumentaci systému převodu najdete na *Převod e-knihy* (stránka 55)

Kdykoliv předáte parametry **ebook-convert**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

Volby a výchozí hodnoty pro změnu voleb závisí na vstupním i výstupním formátu, takže byste je měli vždy zkontrolovat pomocí:

ebook-convert myfile.input_format myfile.output_format -h

Níže jsou volby, které jsou společné pro všechny převody, následované volbami konkrétními pro každý vstupní a výstupní formát.

- *Volby Vstupu* (stránka 246)
- *Volby Výstupu* (stránka 246)
- *Vzhled* (stránka 247)
- *Heuristické Zpracování* (stránka 249)
- *Najít A Nahradit* (stránka 250)
- *Detekce Struktury* (stránka 250)
- *Obsah* (stránka 251)
- *Metadata* (stránka 251)
- *Ladění* (stránka 252)

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--input-profile
Zadejte vstupní profil. Vstupní profil dává převodnímu systému informaci, jak interpretovat různé informace ve vstupním dokumentu. Například délky závislé na rozlišení (t.j. délky v pixelech). Možnosti jsou:cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes
Seznam vestavěných názvů předpisů. Z vestavěného předpisu můžete vytvořit e-knihu takto: ebook-convert "Nazev Predpisu.recipe" output.epub

--output-profile
Zadejte výstupní profil. Výstupní profil říká převodnímu systému, jak optimalizovat vytvořený dokument pro zadané zařízení. V některých případech může být výstupní profil použit pro optimalizaci výstupu pro konkrétní zařízení, ale je to zřídka potřeba. Možnosti jsou:cybookg3, cybook_opus, default, generic_eink, gene-

ric_eink_hd, generic_eink_large, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_pw, kindle_pw3, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version
zobrazit číslo verze programu a ukončit

Volby Vstupu

Volby nastavení zpracování vstupního souboru mobi

--input-encoding
Zadejte kódování znaků vstupního dokumentu. Pokud je tato možnost nastavena, přepíše jakékoliv kódování deklarované samotným dokumentem. Zvláště užitečné pro dokumenty, které nedeklarují kódování nebo mají nesprávně deklarované kódování.

Volby Výstupu

Volby nastavení zpracování výstupního souboru epub

--dont-split-on-page-breaks
Vypnout rozdělování na koncích stránek. Vstupní soubory se obyčejně automaticky rozdělují na každém konci stránky na dva soubory. Tak vznikne výstupní e-kniha, která může být analyzována rychleji a za použití méně prostředků. Rozdělování je však pomalé a pokud zdrojový soubor obsahuje velmi velký počet konců stránek, měli byste rozdělování na koncích stránek vypnout.

--epub-flatten
Tato volba je nutná, pouze pokud máte v úmyslu použít EPUB v FBReaderJ. Urovná to souborový systém uvnitř EPUB, všechny soubory dá na nejvyšší úroveň.

--epub-inline-toc
Vložit vložený obsah, který se objeví jako část hlavního obsahu knihy.

--epub-toc-at-end
Dát vložený obsah na konec knihy místo na začátek.

--extract-to
Rozbalit obsah vygenerovaného souboru EPUB do zadaného adresáře. Obsah adresáře je nejdříve odstraněn, takže buď te opatrní.

--flow-size
Split all HTML files larger than this size (in KB). This is necessary as most EPUB readers cannot handle large file sizes. The default of 260KB is the size required for Adobe Digital Editions. Set to 0 to disable size based splitting.

--no-default-epub-cover
Pokud vstupní soubor nemá žádnou obálku a vy jste žádnou nezadali, je obyčejně vygenerována výchozí obálka s názvem, autorem atd. Tato volba zakáže generování této obálky.

--no-svg-cover
Nepoužívat SVG pro obálku knihy. Tuto možnost použijte, pokud vaše soubory EPUB budou používány v zařízení, které nepodporuje SVG, jako je iPhone nebo JetBook Lite. Bez této volby budou taková zařízení zobrazovat obálku jako prázdnou stránku.

--preserve-cover-aspect-ratio
Při použití obálky SVG tato volba způsobí změnu velikosti obálky, aby pokryla dostupné oblasti obrazovky, ale přesto zachová její poměr stran (poměr šířky k výšce). To znamená, že mohou být vidět bílé okraje po stranách

nebo v horní a dolní části obrázku, ale obrázek nebude nikdy zdeformovaný. Bez této volby může být obrázek mírně deformovaný, ale nebudou vidět žádné okraje.

--pretty-print
Pokud je zadáno, výstupní modul se pokusí vytvořit výstup, který je co nejvíce čitelný pro člověka. Nemusí to mít žádný vliv na některé výstupní moduly.

--toc-title
Název pro vygenerovaný vložený obsah.

Vzhled

Volby nastavení vzheldu výstupu

--asciiize
Překódovat znaky unicode do ASCII. Používejte opatrně, protože to nahradí znaky unicode za ASCII. Například to nahradí "Михаил Горбачёв" za "Michail Gorbačov". Také pamatujte, že v případech, kdy je více zastoupení znaku (například znaky společné pro čínštinu a japonštinu), bude použito zastoupení podle aktuálního jazyku rozhraní Calibre.

--base-font-size
Základní velikost písma v bodech. Všechny velikosti písma ve vytvořené knize budou mít upravené měřítko na základě této velikosti. Výběrem větší velikosti můžete zvětšit písma ve výstupu a naopak. Ve výchozím nastavení je základní velikost písma zvolena na základě vámi zvoleného výstupního profilu.

--change-justification
Změnit zarovnání textu. Hodnota "doleva" převede všechen text zarovnaný do bloku ve zdroji na text zarovnaný vlevo (tj. nezarovnaný). Hodnota "do bloku" převede všechen nezarovnaný text na zarovnaný. Hodnota "původní" (výchozí) nezmění zarovnání ve zdrojovém souboru. Pamatujte, že pouze některé výstupní formáty podporují zarovnání.

--disable-font-rescaling
Zakázat všechny úpravy měřítka velikostí písma.

--embed-all-fonts
Vložit každé písmo, na které se odkazuje ve vstupním dokumentu, ale ještě není vloženo. Vyhledá písma ve vašem systému, a pokud je nalezen, budou vložena. Vkládání bude fungovat, pouze pokud formát, do kterého převádíte, podporuje vložená písma, jako jsou EPUB, AZW3, DOCX nebo PDF. Ujistěte se, že máte správnou licenci pro vkládání písem použitých v tomto dokumentu.

--embed-font-family
Vložit zadanou rodinu písem do knihy. To určuje "základní" písmo použité pro knihu. Pokud vstupní dokument určuje svá vlastní písma, mohou přepsat toto základní písmo. Můžete použít volbu filtrování stylu informací pro odebrání písem ze vstupního dokumentu. Pamatujte, že vkládání písem funguje pouze u některých výstupních formátů, hlavně EPUB a AZW3 a DOCX.

--expand-css
Ve výchozím nastavení použije Calibre zjednodušený formulář pro různé vlastnosti css, jako jsou okraje, odsazení, rámeček atd. Tato volba způsobí, že se místo toho použije rozšířený formulář. Pamatujte, že CSS je vždy rozšířené při generování souborů EPUB s výstupním profilem nastaveným na jeden z profilů Nook, protože Nook neovládá zjednodušené CSS.

--extra-css
Buď cesta k šabloně stylů CSS nebo nezpracované CSS. Toto CSS bude připojeno k pravidlům stylu ze zdrojového souboru, takže může být použito k přepsání těchto pravidel.

--filter-css
Čárkami oddělený seznam vlastností CSS, které budou odebrány ze všech pravidel stylů CSS. To je uži-

tečné, pokud přítomnost některých informací o stylu zabraňuje přepsání ve vašem zařízení. Například: font-family,color,margin-left,margin-right

--font-size-mapping
Mapování z názvů písem CSS na velikosti písem v bodech. Příklad nastavení je 12,12,14,16,18,20,22,24. Toto jsou mapování pro velikosti od xx-small do xx-large, s poslední velikostí pro velice velká písma. Algoritmus úpravy měřítka písma používá tyto velikosti pro chytrou úpravu měřítka písem. Výchozí je použití mapování na základě vámi zvoleného výstupního profilu.

--insert-blank-line
Vložit prázdný řádek mezi odstavce. Nebude fungovat, pokud zdrojový soubor nepoužívá odstavce (značky <p> nebo).

--insert-blank-line-size
Nastavit výšku vložených prázdných řádků (v em). Výška řádků mezi odstavci bude dvojnásobek zde nastavené hodnoty.

--keep-ligatures
Zachovat ligatury přítomné ve vstupním dokumentu. Ligatura je speciální vykreslení páru znaků, jako ff, fi, fl atd. Většina čteček nemá podporu pro ligatury ve svých základních písmech, takže je nepravděpodobné, že je správně vykreslí. Ve výchozím nastavení Calibre změní ligaturu na odpovídající pár normálních znaků. Tato volba je namísto toho ponechá.

--line-height
Výška řádku v bodech. Určuje mezery mezi za sebou následujícími řádky textu. Platí pouze pro prvky, které nedefinují svou vlastní výšku řádku. Ve většině případů je užitečnější volba minimální výška řádku. Ve výchozím nastavení se neprovádí žádné zpracování výšky řádku.

--linearize-tables
Některé chybně navržené dokumenty používají tabulky pro ovládání rozložení textu na stránce. Při převodu mají tyto dokumenty často text, který vybíhá mimo stránku a jiné artefakty. Tato volba extrahuje obsah z tabulek a zobrazí ho v linární tvaru.

--margin-bottom
Nastavit dolní okraj v bodech. Výchozí je 5.0. Nastavení na hodnotu menší než nula způsobí, že se nenastaví žádné okraje (bude zachováno nastavení okrajů v původním dokumentu). Poznámka: 72 bodů je 1 palec

--margin-left
Nastavit levý okraj v bodech. Výchozí je 5.0. Nastavení na hodnotu menší než nula způsobí, že se nenastaví žádné okraje (bude zachováno nastavení okrajů v původním dokumentu). Poznámka: 72 bodů je 1 palec

--margin-right
Nastavit pravý okraj v bodech. Výchozí je 5.0. Nastavení na hodnotu menší než nula způsobí, že se nenastaví žádné okraje (bude zachováno nastavení okrajů v původním dokumentu). Poznámka: 72 bodů je 1 palec

--margin-top
Nastavit horní okraj v bodech. Výchozí je 5.0. Nastavení na hodnotu menší než nula způsobí, že se nenastaví žádné okraje (bude zachováno nastavení okrajů v původním dokumentu). Poznámka: 72 bodů je 1 palec

--minimum-line-height
Minimální výška řádku jako procento vypočítané velikosti písma prvku. Calibre zajistí, že každý prvek bude mít výšku řádku alespoň podle tohoto nastavení bez ohledu na to, co určuje vstupní dokument. Nastavte na nulu pro zakázání. Výchozí je 120 %. Použijte toto nastavení přednostně pro přímé určení výšky řádku, pokud nevíte, co děláte. Například můžete dosáhnout textu s dvojitým řádkováním nastavením na 240.

--remove-paragraph-spacing
Odebrat mezery mezi odstavci. Také nastaví odsazení odstavce na 1,5 em. Odebrání mezer nebude fungovat, pokud zdrojový soubor nepoužívá odstavce (značky <p> nebo).

--remove-paragraph-spacing-indent-size
Když Calibre odebírá prázdné řádky mezi odstavci, automaticky nastaví odsazení odstavce, aby bylo zajištěno snadné rozlišení odstavců. Tato volba určuje šířku tohoto odsazení (v em). Pokud nastavíte zápornou hodnotu, pak bude použito odsazení zadané ve vstupním dokumentu, to znamená, že Calibre nezmění odsazení.

--smarten-punctuation
Převést jednoduché uvozovky, pomlčky a tři tečky na jejich typograficky správné ekvivalenty. Pro podrobnosti přejděte na http://daringfireball.net/projects/smartypants

--subset-embedded-fonts
Podmnožina všech vložených písem. Každé vložené písmo je zmenšeno, aby obsahovalo pouze glyfy použité v tomto dokumentu. To zmenšuje velikost souborů písem. Užitečné, pokud vkládáte obzvlášť velké písmo se spoustou nevyužitých glyfů.

--unsmarten-punctuation
Převést ozdobné uvozovky, pomlčky a tři tečky na jejich prosté ekvivalenty.

Heuristické Zpracování

Změnit text a strukturu dokumentu pomocí běžných vzorů. Ve výchozím nastavení zakázáno. Použijte –enable-heuristics pro povolení. Jednotlivé akce mohou být zakázány pomocí voleb –disable-*.

--disable-dehyphenate
Analyzovat rozdělená slova v celém dokumentu. Samotný dokument je použitý jako slovník pro určení, zda by měly být pomlčky zachovány nebo odebrány.

--disable-delete-blank-paragraphs
Odebrat z dokumentu prázdné odstavce, pokud se vyskytují mezi všemi ostatními odstavci

--disable-fix-indents
Změnit odsazení vytvořená z více entit pevné mezery na odsazení CSS.

--disable-format-scene-breaks
Vlevo zarovnané značky přerušení příběhu jsou zarovnané na střed. Nahradit měkká přerušení příběhu, která používají více prázdných řádků, vodorovnými čárami.

--disable-italicize-common-cases
Hledat běžná slova a vzory, které označují kurzívu a převést je na kurzívu.

--disable-markup-chapter-headings
Rozpoznat neformátované nadpisy a podnadpisy kapitol. Změnit je na značky h2 a h3. Toto nastavení nevytvoří obsah, ale může být použito společně s detekcí struktury pro jeho vytvoření.

--disable-renumber-headings
Hledá výskyty po sobě následujících značek <h1> nebo <h2>. Značky jsou přečíslovány, aby se zabránilo rozdělení uprostřed nadpisů kapitol.

--disable-unwrap-lines
Zrušit zalomení řádků pomocí interpunkce a jiných vodítek formátování.

--enable-heuristics
Povolit heuristické zpracování. Tato volba musí být nastavena, aby se provedlo jakékoliv heuristické zpracování.

--html-unwrap-factor
Měřítko použité k určení délky, při které by mělo být řádku zrušeno zalomení. Platné hodnoty jsou desetinná čísla mezi 0 a 1. Výchozí je 0,4, těsně pod mediánem délky řádku. Pokud by pouze pár řádků v dokumentu vyžadovalo zrušení zalomení, měla by být tato hodnota zmenšena.

--replace-scene-breaks
Nahradit přerušení příběhu zadaným textem. Ve výchozím nastavení je použit text ze vstupního dokumentu.

Najít A Nahradit

Změnit text a strukturu dokumentu pomocí vzorů definovaných uživatelem.

--search-replace
Cesta k souboru obsahující regulární výrazy hledávání a nahrazování. Soubor musí obsahovat střídavě řádky regulárního výrazu následované vzorem nahrazení (což může být prázdný řádek). Regulární výraz musí mít syntaxi Python Regex a soubor musí mít kódování UTF-8.

--sr1-replace
Nahrazení, které má nahradit text nalezený pomocí sr1-search.

--sr1-search
Hledaný vzor (regulární výraz), který má být nahrazen pomocí sr1-replace.

--sr2-replace
Nahrazení, které má nahradit text nalezený pomocí sr2-search.

--sr2-search
Hledaný vzor (regulární výraz), který má být nahrazen pomocí sr2-replace.

--sr3-replace
Nahrazení, které má nahradit text nalezený pomocí sr2-search.

--sr3-search
Hledaný vzor (regulární výraz), který má být nahrazen pomocí sr3-replace.

Detekce Struktury

Nastavení autodetekce struktury dokumentu.

--chapter
Výraz XPath pro rozpoznání nadpisů kapitol. Výchozí nastavení je považovat značky <h1> nebo <h2> obsahující slova “chapter” (kapitola), “book”(kniha), “section” (oddíl), “prologue” (prolog), “epilogue” (epilog) nebo “part” (část) za nadpisy kapitol, stejně jako jakékoliv značky s class=”chapter” (třída kapitola). Použitý výraz musí vyhodnotit seznam prvků. Pro zakázání rozpoznávání kapitol použijte výraz “/”. Další nápovědu k používání této funkce naleznete v kurzu XPath v uživatelské příručce Calibre.

--chapter-mark
Zadejte, jak označit rozpoznané kapitoly. Hodnota “konecstránky” vloží před kapitoly konce stránek. Hodnota “čára” vloží před kapitoly čáru. Hodnota “žádné” zakáže označování kapitol a hodnota “obojí” použije pro označení kapitol jak konce stránek, tak čáry.

--disable-remove-fake-margins
Některé dokumenty zadávají okraje stránek zadáním levého a pravého okraje každému jednotlivému odstavci. Calibre se pokusí rozpoznat a odebrat tyto okraje. Někdy to může způsobit odebírání okrajů, které neměly být odebrány. V tomto případě můžete zakázat odebírání.

--insert-metadata
Vložit metadata knihy na začátek knihy. To je užitečné, pokud vaše čtečka e-knih nepodporuje zobrazení nebo hledání metadat přímo.

--page-breaks-before
Výraz XPath. Konce stránek jsou vloženy před zadané prvky. Pro zakázání použijte výraz: /

--prefer-metadata-cover
Použít obálku rozpoznanou ve zdrojovém souboru přednostně před zadanou obálkou.

--remove-first-image
Odebrat první obrázek ze vstupní e-knihy. Užitečné, pokud má vstupní dokument obrázek obálky, který není

identifikován jako obálka. Pokud v tomto případě nastavíte obálku v Calibre, výstupní dokument bude mít dva obrázky obálky, pokud nezadáte tuto volbu.

--start-reading-at
Výraz XPath pro rozpoznání místa v dokumentu, na kterém začít číst. Některé programy pro čtení e-knih (hlavně Kindle) používají toto místo jako pozici, na které se otevře kniha. Další nápovědu k používání této funkce naleznete v kurzu XPath v uživatelské příručce Calibre.

Obsah

Nastavení automatického generování obsahu. Pokud má zdrojový soubor obsah, bude ve výchozím nastavení použit přednostně před automaticky generovaným.

--duplicate-links-in-toc
Při vytváření obsah z odkazů ve vstupním dokumentu povolit duplicitní položky, tj. povolit více než jednu položku se stejným textem, za předpokladu že odkazují na rozdílná umístění.

--level1-toc
Výraz XPath, který určuje všechny značky, které by měly být přidány do obsahu na první úroveň. Pokud je toto zadáno, má to přednost před jinými formami automatického rozpoznávání. Příklady naleznete v kurzu XPath v uživatelské příručce Calibre.

--level2-toc
Výraz XPath, který určuje všechny značky, které by měly být přidány do obsahu na druhou úroveň. Každá položka je přidána pod předchozí položku první úrovně. Příklady naleznete v kurzu XPath v uživatelské příručce Calibre.

--level3-toc
Výraz XPath, který určuje všechny značky, které by měly být přidány do obsahu na třetí úroveň. Každá položka je přidána pod předchozí položku druhé úrovně. Příklady naleznete v kurzu XPath v uživatelské příručce Calibre.

--max-toc-links
Maximální počet odkazů vložených do obsahu. Nastavte 0 pro zakázání. Výchozí je: 50. Odkazy jsou přidány do obsahu, pouze pokud bylo rozpoznáno méně kapitol, než je mezní hodnota.

--no-chapters-in-toc
Nepřidávat automaticky rozpoznané kapitoly do obsahu.

--toc-filter
Odebrat z obsahu položky, jejichž názvy odpovídají zadanému regulárnímu výrazu. Odpovídající položky a všechny jejich podřízené položky budou odebrány.

--toc-threshold
Pokud je rozpoznáno méně než tento počet kapitol, budou odkazy přidány do obsahu. Výchozí: 6

--use-auto-toc
Pokud má již zdrojový soubor obsah, je obyčejně použit přednostně před automaticky generovaným. Pomocí tohoto nastavení je vždy použit ten automaticky generovaný.

Metadata

Volby pro nastavení metadat ve výstupu

--author-sort
Řetězec použitý při řazení podle autora.

--authors
Nastavit autory. Více autorů by mělo být odděleno znaky "&".

--book-producer
Nastavit nakladatele knihy.

--comments
Nastavit popis e-knihy.

--cover
Nastavit obálku na zadaný soubor nebo URL

--isbn
Nastavit ISBN knihy.

--language
Nastavit jazyk.

--pubdate
Nastavit datum vydání.

--publisher
Nastavit vydavatele e-knihy.

--rating
Nastavit hodnocení. Mělo by to být číslo mezi 1 a 5.

--read-metadata-from-opf, -m
Číst metadata ze zadaného souboru OPF. Metadata načtená z tohoto souboru přepíší jakákoliv metadata ve zdrojovém souboru.

--series
Nastavit sérii, do které tato e-kniha patří.

--series-index
Nastavit pořadí knihy v této sérii.

--tags
Nastavit štítky pro knihu. Měl by to být seznam oddělený čárkami.

--timestamp
Nastavit časové razítko knihy (nikde se již nepoužívá)

--title
Nastavit název.

--title-sort
Verze názvu použitá při řazení.

Ladění

Volby pomáhající s laděním převodu

--debug-pipeline, -d
Uložit výstup z rozdílných fází převodního řetězce do zadaného adresáře. Užitečné, pokud si nejste jisti, v jaké fázi zpracování převodu se objevuje chyba.

--verbose, -v
Úroveň podrobnosti. Zadejte vícekrát pro větší podrobnost. Zadání dvakrát bude mít za následek úplnou podrobnost, jednou střední podrobnost a nulakrát nejmenší podrobnost.

ebook-edit

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

Spustit nástroj Calibre pro úpravu knihy. Volitelně můžete také zadat názvy souborů uvnitř knihy, které budou automaticky otevřeny pro úpravu.

Kdykoliv předáte parametry **ebook-edit**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--detach
: Odpojit od řídícího terminálu, pokud nějaký je (pouze pro Linux)

--help, -h
: zobrazit tuto zprávu nápovědy a ukončit

--version
: zobrazit číslo verze programu a ukončit

ebook-meta

```
ebook-meta ebook_file [volby]
```

Číst/Zapisovat metadata z/do souborů e-knih.

Podporované formáty pro čtení metadat: azw, azw1, azw3, azw4, cbr, cbz, chm, docx, epub, fb2, html, htmlz, imp, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Podporované formáty pro zápis metadat: azw, azw1, azw3, azw4, docx, epub, fb2, htmlz, lrf, mobi, pdb, pdf, prc, rtf, tpz, txtz

Různé typy souborů podporují různé druhy metadat. Pokud se pokusíte nastavit nějaká metadata k typu souboru, který je nepodporuje, metadata budou bez upozornění ignorována.

Kdykoliv předáte parametry **ebook-meta**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--author-sort
: Řetězec použitý při řazení podle autora. Pokud není zadán a je zadán autor, bude automaticky generován z autora.

--authors, -a
: Nastavit autory. Více autorů by mělo být odděleno znakem &. Jména autorů by měla být v pořadí křestní jméno, příjmení.

--book-producer, -k
: Nastavit nakladatele knihy.

--category
: Nastavit kategorii knihy.

--comments, -c
: Nastavit popis e-knihy.

--cover
Nastavit obálku zadaného souboru.

--date, -d
Nastavit datum vydání.

--from-opf
Načíst metadata ze zadaného souboru OPF a použít je k nastavení metadat v e-knize. Metadata zadaná v příkazovém řádku přepíšou metadata načtená ze souboru OPF.

--get-cover
Získat obálku z e-knihy a uložit ji jako zadaný soubor.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--identifier
Nastavit identifikátory pro knihu, může být zadáno vícekrát. Například: *--identifier* (stránka 254) uri:http://acme.com *--identifier* (stránka 254) isbn:12345. Pro odebrání identifikátoru zadejte prázdnou hodnotu, *--identifier* (stránka 254) isbn:. Pamatujte, že identifikátor označený jako identifikátor balíčku nemůže být u souborů EPUB odebrán.

--index, -i
Nastavit pořadí knihy v této sérii.

--isbn
Nastavit ISBN knihy.

--language, -l
Nastavit jazyk.

--lrf-bookid
Nastavit BookID v souborech LRF

--publisher, -p
Nastavit vydavatele e-knihy.

--rating, -r
Nastavit hodnocení. Mělo by to být číslo mezi 1 a 5.

--series, -s
Nastavit sérii, do které tato e-kniha patří.

--tags
Nastavit štítky pro knihu. Měl by to být seznam oddělený čárkami.

--title, -t
Nastavit název.

--title-sort
Verze názvu použitá pro řazení. Pokud není zadán a je zadán název, bude automaticky generován z názvu.

--to-opf
Zadejte název souboru OPF. Metadata budou zapsána do souboru OPF.

--version
zobrazit číslo verze programu a ukončit

ebook-polish

```
ebook-polish [options] input_file [output_file]
```

Vylepšení knih je celé o dotažení vašich pečlivě vytvářených e-knih k dokonalosti.

Vylepšení se pokouší minimalizovat změny vnitřního kódu vaší e-knihy. Na rozdíl od převodu nedělá následující: neuhlazuje CSS, nepřejmenovává soubory, nemění velikosti písma, neupravuje okraje atd. Každá akce provádí pouze minimální sadu změn potřebnou pro požadovaný efekt.

Tento nástroj byste měli použít jako poslední krok při vytváření e-knihy.

Pamatujte, že vylepšení funguje pouze u souborů ve formátech AZW3neboEPUB.

Kdykoliv předáte parametry **ebook-polish**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--compress-images, -i
: Losslessly compress images in the book, to reduce the filesize, without affecting image quality.

--cover, -c
: Cesta k obrázku obálky. Mění obálku zadanou v e-knize. Pokud není přítomna žádná obálka, nebo pokud není obálka správně identifikována, vloží novou obálku.

--embed-fonts, -e
: Vložit všechna písma, na která je odkazováno v dokumentu a nejsou již vložena. To vyhledá písma ve vašem počítači, a pokud jsou nalezena, budou vložena do dokumentu. Ujistěte se, že máte řádnou licenci pro vkládání písem použitých v tomto dokumentu.

--help, -h
: zobrazit tuto zprávu nápovědy a ukončit

--jacket, -j
: Vložit na začátek knihy stránku "přebalu knihy", která obsahuje všechna metadata knihy, jako jsou název, štítky, autoři, série, komentáře atd. Jakýkoliv předchozí přebal knihy bude nahrazen.

--opf, -o
: Cesta k souboru OPF. Metadata v knize jsou aktualizována ze souboru OPF.

--remove-jacket
: Odebrat dříve vloženou stránku přebalu knihy.

--remove-unused-css, -u
: Odebrat všechna nepoužívaná pravidla CSS ze šablon stylů a značek <style>. Některé knihy vytvořené pomocí výrobních šablon mohou mít velký počet dodatečných pravidel CSS, která neodpovídají skutečnému obsahu. Tato dodatečná pravidla mohou zpomalovat čtečky, které je musí všechny analyzovat.

--smarten-punctuation, -p
: Převést pomlčky, tři tečky, uvozovky, vícenásobné pomlčky atd. v prostém textu na jejich typograficky správné ekvivalenty. Pamatujte, že algoritmus může někdy generovat nesprávné výsledky, obzvláště pokud se to týká jednoduchých uvozovek na začátku zkrácených tvarů.

--subset-fonts, -f
: Částečné vkládání písem znamená zmenšení vloženého písma, aby obsahovalo pouze znaky použité z tohoto písma v knize. To značně zmenšuje velikost souborů písem (poloviční velikosti souborů písem jsou běžné). Například pokud kniha používá konkrétní písmo pro nadpisy, pak částečné vkládání zmenší toto písmo, aby obsahovalo pouze znaky přítomné v nadpisech v knize. Nebo pokud kniha vloží verze písma pro tučné písmo a kurzívu, ale texty tučným písmem a kurzívou jsou poměrně vzácné nebo úplně chybí, pak může být tučné písmo

a kurzíva buď zmenšeno, nebo úplně odebráno. Jediná nevýhoda částečného vkládání písem je ta, že pokud se později rozhodnete přidat další text do svých knih, nově přidaný text nemusí být pokryt podmnožinou písma.

--verbose
Vytváří podrobnější výstup, užitečné pro ladění.

--version
zobrazit číslo verze programu a ukončit

ebook-viewer

```
ebook-viewer [options] file
```

Zobrazit e-knihu.

Kdykoliv předáte parametry **ebook-viewer**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--continue
Pokračovat ve čtení dříve otevřené knihy

--debug-javascript
Vypsat výstraha javascriptu a zprávy konzole do konzole

--detach
Odpojit od řídícího terminálu, pokud nějaký je (pouze pro Linux)

--full-screen, -f
Pokud je zadáno, okno prohlížeče se pokusí při spuštění otevřít na celou obrazovku.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--open-at
Pozice, na které otevřít zadanou knihu. Pozice je umístění zobrazené v levém horním rohu prohlížeče.

--raise-window
Pokud je zadáno, okno prohlížeče se pokusí při spuštění zobrazit v popředí.

--version
zobrazit číslo verze programu a ukončit

fetch-ebook-metadata

```
fetch-ebook-metadata [options]
```

Načíst metadata knihy z online zdrojů. Musíte zadat alespoň název, autory nebo ISBN.

Kdykoliv předáte parametry **fetch-ebook-metadata**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--authors, -a
Autoři knihy

--cover, -c
Zadejte název souboru. Pokud existuje obálka, bude do něj uložena. Bez této volby nebude stažena žádná obálka.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--isbn, -i
ISBN knihy

--opf, -o
Výstup metadat ve formátu OPF namísto textu čitelného pro člověka.

--timeout, -d
Časový limit v sekundách. Výchozí je 30

--title, -t
Název knihy

--verbose, -v
Vypsat protokol do konzole (stderr)

--version
zobrazit číslo verze programu a ukončit

lrf2lrs

```
lrf2lrs book.lrf
```

Převést soubor LRF na soubor LRS (XML s kódováním UTF-8)

Kdykoliv předáte parametry **lrf2lrs**, které obsahují mezery, uzavřete parametry uvozovkami. Například “C:nějaká cesta s mezerami”

[options]

--dont-output-resources
Neukládat vložené soubory obrázků a písem na disk

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--output, -o
Výstupní soubor LRS

--verbose
Více podrobností

--version
zobrazit číslo verze programu a ukončit

lrfviewer

```
lrfviewer [options] book.lrf
```

Číst e-knihu LRF book.lrf

Kdykoliv předáte parametry **lrfviewer**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--disable-hyphenation
Zakázat dělení slov. Mělo by to výrazně zrychlit vykreslování.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--profile
Profilovat nástroj pro vykreslování LRF

--verbose
Vypsat více informací o procesu vykreslování

--version
zobrazit číslo verze programu a ukončit

--visual-debug
Zapnout vizuální pomůcky pro ladění modulu vykreslování

--white-background
Ve výchozím nastavení je pozadí skoro bílé, protože je to snadnější pro oči. Použijte tuto volbu, aby bylo pozadí čistě bílé.

lrs2lrf

```
lrs2lrf [options] file.lrs
```

Kompilovat soubor LRS na soubor LRF.

Kdykoliv předáte parametry **lrs2lrf**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--lrs
Převést LRS na LRS, užitečné pro ladění.

--output, -o
Cesta k výstupnímu souboru

--verbose
Podrobné zpracování

--version
zobrazit číslo verze programu a ukončit

web2disk

```
web2disk URL
```

Kde URL je například http://google.com

Kdykoliv předáte parametry **web2disk**, které obsahují mezery, uzavřete parametry uvozovkami. Například "C:nějaká cesta s mezerami"

[options]

--base-dir, -d
Základní adresář, do kterého je uložena URL. Výchozí je .

--delay
Minimální interval mezi za sebou následujícími načteními v sekundách. Výchozí je 0 s

--dont-download-stylesheets
Nestahovat šablony stylů CSS.

--encoding
Kódování znaků webových stránek, které se pokoušíte stáhnout. Výchozí je pokusit se odhadnout kódování.

--filter-regexp
Jakýkoliv odkaz, který odpovídá tomuto regulárnímu výrazu, bude ignorován. Tato volba může být zadána vícekrát, v tom případě dokud kterýkoliv regulární výraz odpovídá odkazu, bude ignorován. Pokud je zadáno filtrování regulárních výrazů a porovnání regulárních výrazů, pak filtrování regulárních výrazů je použito první.

--help, -h
zobrazit tuto zprávu nápovědy a ukončit

--match-regexp
Pouze odkazy, které odpovídají tomuto regulárnímu výrazu, budou následovány. Tato volba může být zadána vícekrát, v tom případě bude odkaz následován, dokud odpovídá kterémukoliv regulárnímu výrazu. Ve výchozím nastavení jsou následovány všechny odkazy.

--max-files, -n
Maximální počet stahovaných souborů. Toto se použije pouze na soubory v <a href>štítcích. Výchozí je 9223372036854775807

--max-recursions, -r
Maximální počet úrovní rekurze, tj. hloubka sledovaných odkazů. Výchozí 1

--timeout, -t
Časový limit čekání na odezvu ze serveru v sekundách. Výchozí: 10.0 s

--verbose
Zobrazit podrobné výstupní informace. Užitečné pro ladění

--version
zobrazit číslo verze programu a ukončit

1.11.2 Nedokumentované příkazy

- ebook-device
- markdown-calibre

Použití pro nedokumentované příkazy můžete vidět jejich provedením bez parametrů v terminálu.

1.12 Nastavení vývojového prostředí calibre

calibre is completely open source, licensed under the GNU GPL v3[93]. This means that you are free to download and modify the program to your heart's content. In this section, you will learn how to get a calibre development environment set up on the operating system of your choice. calibre is written primarily in Python[94] with some C/C++ code for speed and system interfacing. Note that calibre is not compatible with Python 3 and requires at least Python 2.7.9.

Obsah

- *Design philosophy* (stránka 260)
  - *Rozvržení kódu* (stránka 260)
- *Getting the code* (stránka 261)
  - *Submitting your changes to be included* (stránka 262)
- *Windows development environment* (stránka 263)
- *OS X development environment* (stránka 263)
- *Linux development environment* (stránka 264)
- *Having separate "normal" and "development" calibre installs on the same computer* (stránka 264)
- *Debugging tips* (stránka 265)
  - *Using print statements* (stránka 265)
  - *Using an interactive python interpreter* (stránka 265)
  - *Using the python debugger as a remote debugger* (stránka 265)
  - *Používání debuggeru ve vašem oblíbeném Python IDE* (stránka 266)
  - *Spuštení libovolých skriptů uvnitř python prostření Calibre* (stránka 266)
- *Using calibre in your projects* (stránka 266)
  - *Binary install of calibre* (stránka 266)
  - *Source install on Linux* (stránka 267)
- *API documentation for various parts of calibre* (stránka 267)

1.12.1 Design philosophy

calibre has its roots in the Unix world, which means that its design is highly modular. The modules interact with each other via well defined interfaces. This makes adding new features and fixing bugs in calibre very easy, resulting in a frenetic pace of development. Because of its roots, calibre has a comprehensive command line interface for all its functions, documented in `generated/en/cli-index`.

The modular design of calibre is expressed via `Plugins`. There is a *tutorial* (stránka 193) on writing calibre plugins. For example, adding support for a new device to calibre typically involves writing less than a 100 lines of code in the form of a device driver plugin. You can browse the built-in drivers[95]. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (stránka 23) for fetching news. For more examples of plugins designed to add features to calibre, see the plugin index[96].

Rozvržení kódu

Všechen kód Calibre napsaný v python-u je v balíčku `calibre`. Tento balíček obsahují následující hlavní subbalíčky.

[93] http://www.gnu.org/copyleft/gpl.html
[94] https://www.python.org
[95] https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices
[96] http://www.mobileread.com/forums/showthread.php?p=1362767#post1362767

- devices - Všechny ovládače od zařízení. Náhledněte do některého z vestaněných ovládačů pro získání informaci jak ovladače pracují.
  - Pro více podrobnosti viz `devices.interface`, který definuje rozhraní podporované obladači zařízení a `devices.usbms` , který definuje obecný - generic - ovládač, který připojuje USBMS zařízení. Všechny Calibre ovládače postavené na USBMS z něj dědí.
- ebooks - Kódy týkající se převodu a metadat ebooků. Vhodným výchozím bodem je balíček `calibre.ebooks.conversion.cli`, který pohání příkaz **`ebook-convert`**. Proces převodu je ovládán skrze `conversion.plumber`. Kódy nezávislé na formátu jsou v balíčku `ebooks.oeb`, naopak kódy závislé na formátu se nacházejí v `ebooks.format_name`.
  - Čtení, zápis a stahování metadat je v balíčku `ebooks.metadata`
  - Conversion happens in a pipeline, for the structure of the pipeline, see *Úvod* (stránka 56). The pipeline consists of an input plugin, various transforms and an output plugin. The code that constructs and drives the pipeline is in `plumber.py`. The pipeline works on a representation of an ebook that is like an unzipped epub, with manifest, spine, toc, guide, html content, etc. The class that manages this representation is OEBBook in `ebooks.oeb.base`. The various transformations that are applied to the book during conversions live in `oeb/transforms/*.py`. And the input and output plugins live in `conversion/plugins/*.py`.
  - Ebook editing happens using a different container object. It is documented in *API Documentation for the ebook editing tools* (stránka 273).
- db - The database back-end. See *Dokumentace API pro rozhraní databáze* (stránka 267) for the interface to the calibre library.
- content server: `library.server` is the calibre Content Server.
- gui2 - The Graphical User Interface. GUI initialization happens in `gui2.main` and `gui2.ui`. The ebook-viewer is in `gui2.viewer`. The ebook editor is in `gui2.tweak_book`.

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in linux.py[97].

If you need help understanding the code, post in the development forum[98] and you will most likely get help from one of calibre's many developers.

1.12.2 Getting the code

You can get the calibre source code in two ways, using a version control system or directly downloading a tarball[99].

calibre uses Git[100], a distributed version control system. Git is available on all the platforms calibre supports. After installing Git, you can get the calibre source code with the command:

```
git clone git://github.com/kovidgoyal/calibre.git
```

On Windows you will need the complete path name, that will be something like `C:\Program Files\Git\git.exe`.

calibre is a very large project with a very long source control history, so the above can take a while (10 mins to an hour depending on your internet speed).

If you want to get the code faster, the source code for the latest release is always available as an archive[101].

[97] https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py
[98] http://www.mobileread.com/forums/forumdisplay.php?f=240
[99] http://code.calibre-ebook.com/dist/src
[100] http://www.git-scm.com/
[101] http://code.calibre-ebook.com/dist/src

To update a branch to the latest code, use the command:

```
git pull --no-edit
```

Submitting your changes to be included

If you only plan to make a few small changes, you can make your changes and create a "merge directive" which you can then attach to a ticket in the calibre bug tracker[102]. To do this, make your changes, then run:

```
git commit -am "Comment describing your changes"
git format-patch origin/master --stdout > my-changes
```

This will create a `my-changes` file in the current directory, simply attach that to a ticket on the calibre bug tracker[103]. Note that this will include *all* the commits you have made. If you only want to send some commits, you have to change `origin/master` above. To send only the last commit, use:

```
git format-patch HEAD~1 --stdout > my-changes
```

To send the last *n* commits, replace *1* with *n*, for example, for the last 3 commits:

```
git format-patch HEAD~3 --stdout > my-changes
```

Be careful to not include merges when using `HEAD~n`.

If you plan to do a lot of development on calibre, then the best method is to create a GitHub[104] account. Below is a basic guide to setting up your own fork of calibre in a way that will allow you to submit pull requests for inclusion into the main calibre repository:

- Setup git on your machine as described in this article: Setup Git[105]
- Setup ssh keys for authentication to GitHub, as described here: Generating SSH keys[106]
- Go to https://github.com/kovidgoyal/calibre and click the *Fork* button.
- In a Terminal do:

```
git clone git@github.com:<username>/calibre.git
git remote add upstream https://github.com/kovidgoyal/calibre.git
```

  Replace <username> above with your github username. That will get your fork checked out locally.

- You can make changes and commit them whenever you like. When you are ready to have your work merged, do a:

```
git push
```

  and go to `https://github.com/<username>/calibre` and click the *Pull Request* button to generate a pull request that can be merged.

- You can update your local copy with code from the main repo at any time by doing:

```
git pull upstream
```

You should also keep an eye on the calibre development forum[107]. Before making major changes, you should discuss them in the forum or contact Kovid directly (his email address is all over the source code).

[102] https://bugs.launchpad.net/calibre
[103] https://bugs.launchpad.net/calibre
[104] https://github.com
[105] https://help.github.com/articles/set-up-git
[106] https://help.github.com/articles/generating-ssh-keys
[107] http://www.mobileread.com/forums/forumdisplay.php?f=240

1.12.3 Windows development environment

Poznámka: You must also get the calibre source code separately as described above.

Install calibre normally, using the Windows installer. Then open a Command Prompt and change to the previously checked out calibre code directory. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre is the directory that contains the src and resources sub-directories.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src directory. So, following the example above, it would be `C:\Users\kovid\work\calibre\src`. Here is a short guide[108] to setting environment variables on Windows.

Once you have set the environment variable, open a new command prompt and check that it was correctly set by using the command:

```
echo %CALIBRE_DEVELOP_FROM%
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src\calibre\__init__.py` in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **`calibredb`**. The very first line of output should be `Hello, world!`.

You can also setup a calibre development environment inside the free Microsoft Visual Studio, if you like, following the instructions here[109].

1.12.4 OS X development environment

Poznámka: You must also get the calibre source code separately as described above.

Install calibre normally using the provided .dmg. Then open a Terminal and change to the previously checked out calibre code directory, for example:

```
cd /Users/kovid/work/calibre
```

calibre is the directory that contains the src and resources sub-directories. Ensure you have installed the calibre commandline tools via *Preferences->Advanced->Miscellaneous* in the calibre GUI.

The next step is to create a bash script that will set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src directory when running calibre in debug mode.

Create a plain text file:

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

Save this file as `/usr/bin/calibre-develop`, then set its permissions so that it can be executed:

[108] https://docs.python.org/2/using/windows.html#excursus-setting-environment-variables
[109] http://www.mobileread.com/forums/showthread.php?t=251201

```
chmod +x /usr/bin/calibre-develop
```

Once you have done this, run:

```
calibre-develop
```

You should see some diagnostic information in the Terminal window as calibre starts up, and you should see an asterisk after the version number in the GUI window, indicating that you are running from source.

1.12.5 Linux development environment

Poznámka: You must also get the calibre source code separately as described above.

calibre is primarily developed on Linux. You have two choices in setting up the development environment. You can install the calibre binary as normal and use that as a runtime environment to do your development. This approach is similar to that used in Windows and OS X. Alternatively, you can install calibre from source. Instructions for setting up a development environment from source are in the INSTALL file in the source tree. Here we will address using the binary as a runtime, which is the recommended method.

Install calibre using the binary installer. Then open a terminal and change to the previously checked out calibre code directory, for example:

```
cd /home/kovid/work/calibre
```

calibre is the directory that contains the src and resources sub-directories.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src directory. So, following the example above, it would be `/home/kovid/work/calibre/src`. How to set environment variables depends on your Linux distribution and what shell you are using.

Once you have set the environment variable, open a new terminal and check that it was correctly set by using the command:

```
echo $CALIBRE_DEVELOP_FROM
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src/calibre/__init__.py` in your favorite editor and add the line:

```
print ("Hello, world!")
```

near the top of the file. Now run the command **calibredb**. The very first line of output should be `Hello, world!`.

1.12.6 Having separate "normal" and "development" calibre installs on the same computer

The calibre source tree is very stable and rarely breaks, but if you feel the need to run from source on a separate test library and run the released calibre version with your everyday library, you can achieve this easily using .bat files or shell scripts to launch calibre. The example below shows how to do this on Windows using .bat files (the instructions for other platforms are the same, just use a shell script instead of a .bat file)

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

1.12.7 Debugging tips

Python is a dynamically typed language with excellent facilities for introspection. Kovid wrote the core calibre code without once using a debugger. There are many strategies to debug calibre code:

Using print statements

This is Kovid's favorite way to debug. Simply insert print statements at points of interest and run your program in the terminal. For example, you can start the GUI from the terminal as:

```
calibre-debug -g
```

Similarly, you can start the ebook-viewer as:

```
calibre-debug -w /path/to/file/to/be/viewed
```

The ebook-editor can be started as:

```
calibre-debug -t /path/to/be/edited
```

Using an interactive python interpreter

You can insert the following two lines of code to start an interactive python session at that point:

```
from calibre import ipython
ipython(locals())
```

When running from the command line, this will start an interactive Python interpreter with access to all locally defined variables (variables in the local scope). The interactive prompt even has TAB completion for object properties and you can use the various Python facilities for introspection, such as `dir()`, `type()`, `repr()`, etc.

Using the python debugger as a remote debugger

You can use the builtin python debugger (pdb) as a remote debugger from the command line. First, start the remote debugger at the point in the calibre code you are interested in, like this:

```
from calibre.rpdb import set_trace
set_trace()
```

Then run calibre, either as normal, or using one of the calibre-debug commands described in the previous section. Once the above point in the code is reached, calibre will freeze, waiting for the debugger to connect.

Now open a terminal or command prompt and use the following command to start the debugging session:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

You can read about how to use the python debugger in the python stdlib docs for the pdb module[110].

Poznámka: By default, the remote debugger will try to connect on port 4444. You can change it, by passing the port parameter to both the set_trace() and the cli() functions above, like this: `set_trace(port=1234)` and `cli(port=1234)`.

Poznámka: The python debugger cannot handle multiple threads, so you have to call set_trace once per thread, each time with a different port number.

Používání debuggeru ve vašem oblíbeném Python IDE

It is possible to use the builtin debugger in your favorite python IDE, if it supports remote debugging. The first step is to add the calibre src checkout to the `PYTHONPATH` in your IDE. In other words, the directory you set as `CALIBRE_DEVELOP_FROM` above, must also be in the `PYTHONPATH` of your IDE.

Then place the IDE's remote debugger module into the `src` subdirectory of the calibre source code checkout. Add whatever code is needed to launch the remote debugger to calibre at the point of interest, for example in the main function. Then run calibre as normal. Your IDE should now be able to connect to the remote debugger running inside calibre.

Spuštení libovolých skriptů uvnitř python prostření Calibre

The **`calibre-debug`** command provides a couple of handy switches to execute your own code, with access to the calibre modules:

```
calibre-debug -c "some python code"
```

is great for testing a little snippet of code on the command line. It works in the same way as the -c switch to the python interpreter:

```
calibre-debug myscript.py
```

can be used to execute your own Python script. It works in the same way as passing the script to the Python interpreter, except that the calibre environment is fully initialized, so you can use all the calibre code in your script. To use command line arguments with your script, use the form:

```
calibre-debug myscript.py -- --option1 arg1
```

The `--` causes all subsequent arguments to be passed to your script.

1.12.8 Using calibre in your projects

It is possible to directly use calibre functions/code in your Python project. Two ways exist to do this:

Binary install of calibre

If you have a binary install of calibre, you can use the Python interpreter bundled with calibre, like this:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

[110] https://docs.python.org/2/library/pdb.html#debugger-commands

Source install on Linux

In addition to using the above technique, if you do a source install on Linux, you can also directly import calibre, as follows:

```
import init_calibre
import calibre

print calibre.__version__
```

It is essential that you import the init_calibre module before any other calibre modules/packages as it sets up the interpreter to run calibre code.

1.12.9 API documentation for various parts of calibre

Dokumentace API pro rozhraní databáze

This API is thread safe (it uses a multiple reader, single writer locking scheme). You can access this API like this:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

If you are in a calibre plugin that is part of the main calibre GUI, you get access to it like this instead:

```
db = self.gui.current_db.new_api
```

class calibre.db.cache.**Cache**(*backend*)

An in-memory cache of the metadata.db file from a calibre library. This class also serves as a threadsafe API for accessing the database. The in-memory cache is maintained in normal form for maximum performance.

SQLITE is simply used as a way to read and write from metadata.db robustly. All table reading/sorting/searching/caching logic is re-implemented. This was necessary for maximum performance and flexibility.

add_books(*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Add the specified books to the library. Books should be an iterable of 2-tuples, each 2-tuple of the form `(mi, format_map)` where mi is a Metadata object and format_map is a dictionary of the form `{fmt: path_or_stream}`, for example: `{'EPUB': '/path/to/file.epub'}`.

Returns a pair of lists: `ids, duplicates`. `ids` contains the book ids for all newly created books in the database. `duplicates` contains the `(mi, format_map)` for all books that already exist in the database as per the simple duplicate detection heuristic used by *has_book()* (stránka 270).

add_custom_book_data(*name*, *val_map*, *delete_first=False*)

Add data for name where val_map is a map of book_ids to values. If delete_first is True, all previously stored data for name will be removed.

add_format(*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Add a format to the specified book. Return True of the format was added successfully.

Parametry

- **replace** – If True replace existing format, otherwise if the format already exists, return False.
- **run_hooks** – If True, file type plugins are run on the format before and after being added.
- **dbapi** – Internal use only.

all_book_ids(*type=<type 'frozenset'>*)
Frozen set of all known book ids.

all_field_for(*field*, *book_ids*, *default_value=None*)
Same as field_for, except that it operates on multiple books at once

all_field_ids(*name*)
Frozen set of ids for all values in the field `name`.

all_field_names(*field*)
Frozen set of all fields names (should only be used for many-one and many-many fields)

author_data(*author_ids=None*)
Return author data as a dictionary with keys: name, sort, link

If no authors with the specified ids are found an empty dictionary is returned. If author_ids is None, data for all authors is returned.

author_sort_from_authors(*authors*)
Given a list of authors, return the author_sort string for the authors, preferring the author sort associated with the author over the computed string.

books_for_field(*name*, *item_id*)
Return all the books associated with the item identified by `item_id`, where the item belongs to the field `name`.

Returned value is a set of book ids, or the empty set if the item or the field does not exist.

copy_cover_to(*book_id*, *dest*, *use_hardlink=False*)
Copy the cover to the file like object `dest`. Returns False if no cover exists or dest is the same file as the current cover. dest can also be a path in which case the cover is copied to it if and only if the path is different from the current path (taking case sensitivity into account).

copy_format_to(*book_id*, *fmt*, *dest*, *use_hardlink=False*)
Copy the format `fmt` to the file like object `dest`. If the specified format does not exist, raises `NoSuchFormat` error. dest can also be a path, in which case the format is copied to it, iff the path is different from the current path (taking case sensitivity into account).

cover(*book_id*, *as_file=False*, *as_image=False*, *as_path=False*)
Return the cover image or None. By default, returns the cover as a bytestring.

WARNING: Using as_path will copy the cover to a temp file and return the path to the temp file. You should delete the temp file when you are done with it.

Parametry

- **as_file** – If True return the image as an open file object (a SpooledTemporaryFile)
- **as_image** – If True return the image as a QImage object
- **as_path** – If True return the image as a path pointing to a temporary file

data_for_find_identical_books()
Return data that can be used to implement *find_identical_books()* (stránka 269) in a worker process without access to the db. See db.utils for an implementation.

data_for_has_book()
Return data suitable for use in *has_book()* (stránka 270). This can be used for an implementation of *has_book()* (stránka 270) in a worker process without access to the db.

delete_custom_book_data(*name*, *book_ids=()*)
Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

embed_metadata(*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)
Update metadata in all formats of the specified book_ids to current metadata in the database.

fast_field_for(*field_obj*, *book_id*, *default_value=None*)
Same as field_for, except that it avoids the extra lookup to get the field object

field_for(*name*, *book_id*, *default_value=None*)
Return the value of the field `name` for the book identified by `book_id`. If no such book exists or it has no defined value for the field `name` or no such field exists, then `default_value` is returned.

`default_value` is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. `default_value` is used for all custom columns.

The returned value for is_multiple fields are always tuples, even when no values are found (in other words, default_value is ignored). The exception is identifiers for which the returned value is always a dict. The returned tuples are always in link order, that is, the order in which they were created.

field_ids_for(*name*, *book_id*)
Return the ids (as a tuple) for the values that the field `name` has on the book identified by `book_id`. If there are no values, or no such book, or no such field, an empty tuple is returned.

find_identical_books(*mi*, *search_restriction=u''*, *book_ids=None*)
Finds books that have a superset of the authors in mi and the same title (title is fuzzy matched). See also *data_for_find_identical_books()* (stránka 268).

format(*book_id*, *fmt*, *as_file=False*, *as_path=False*, *preserve_filename=False*)
Return the ebook format as a bytestring or *None* if the format doesn't exist, or we don't have permission to write to the ebook file.

Parametry

- **as_file** – If True the ebook format is returned as a file object. Note that the file object is a SpooledTemporaryFile, so if what you want to do is copy the format to another file, use *copy_format_to()* (stránka 268) instead for performance.
- **as_path** – Copies the format file to a temp file and returns the path to the temp file
- **preserve_filename** – If True and returning a path the filename is the same as that used in the library. Note that using this means that repeated calls yield the same temp file (which is re-created each time)

format_abspath(*book_id*, *fmt*)
Return absolute path to the ebook file of format *format*. You should almost never use this, as it breaks the threadsafe promise of this API. Instead use, *copy_format_to()* (stránka 268).

Currently used only in calibredb list, the viewer, edit book, compare_format to original format, open with and the catalogs (via get_data_as_dict()).

Apart from the viewer, open with and edit book, I don't believe any of the others do any file write I/O with the results of this call.

format_hash(*book_id*, *fmt*)
Return the hash of the specified format for the specified book. The kind of hash is backend dependent, but is usually SHA-256.

format_metadata(*book_id*, *fmt*, *allow_cache=True*, *update_db=False*)
Return the path, size and mtime for the specified format for the specified book. You should not use path unless you absolutely have to, since accessing it directly breaks the threadsafe guarantees of this API. Instead use the *copy_format_to()* (stránka 268) method.

Parametry

- **allow_cache** – If `True` cached values are used, otherwise a slow filesystem access is done. The cache values could be out of date if access was performed to the filesystem outside of this API.
- **update_db** – If `True` The max_size field of the database is updates for this book.

formats(*book_id*, *verify_formats=True*)
Return tuple of all formats for the specified book. If verify_formats is True, verifies that the files exist on disk.

get_categories(*sort=u'name'*, *book_ids=None*, *already_fixed=None*, *first_letter_sort=False*)
Used internally to implement the Tag Browser

get_custom_book_data(*name*, *book_ids=()*, *default=None*)
Get data for name. By default returns data for all book_ids, pass in a list of book ids if you only want some data. Returns a map of book_id to values. If a particular value could not be decoded, uses default for it.

get_id_map(*field*)
Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a ValueError is raised.

get_ids_for_custom_book_data(*name*)
Return the set of book ids for which name has data.

get_item_id(*field*, *item_name*)
Return the item id for item_name (case-insensitive)

get_item_ids(*field*, *item_names*)
Return the item id for item_name (case-insensitive)

get_item_name(*field*, *item_id*)
Return the item name for the item specified by item_id in the specified field. See also *get_id_map()* (stránka 270).

get_metadata(*book_id*, *get_cover=False*, *get_user_categories=True*, *cover_as_data=False*)
Return metadata for the book identified by book_id as a `Metadata` object. Note that the list of formats is not verified. If get_cover is True, the cover is returned, either a path to temp file as mi.cover or if cover_as_data is True then as mi.cover_data.

get_next_series_num_for(*series*, *field=u'series'*, *current_indices=False*)
Return the next series index for the specified series, taking into account the various preferences that control next series number generation.

Parametry

- **field** – The series-like field (defaults to the builtin series column)
- **current_indices** – If True, returns a mapping of book_id to current series_index value instead.

get_proxy_metadata(*book_id*)
Like *get_metadata()* (stránka 270) except that it returns a ProxyMetadata object that only reads values from the database on demand. This is much faster than get_metadata when only a small number of fields need to be accessed from the returned metadata object.

get_usage_count_by_id(*field*)
Return a mapping of id to usage count for all values of the specified field, which must be a many-one or many-many field.

has_book(*mi*)
Return True iff the database contains an entry with the same title as the passed in Metadata object. The comparison is case-insensitive. See also *data_for_has_book()* (stránka 268).

has_format(*book_id*, *fmt*)
Return True iff the format exists on disk

has_id(*book_id*)
Return True iff the specified book_id exists in the db

init()
Initialize this cache with data from the backend.

multisort(*fields*, *ids_to_sort=None*, *virtual_fields=None*)
Return a list of sorted book ids. If ids_to_sort is None, all book ids are returned.

fields must be a list of 2-tuples of the form (field_name, ascending=True or False). The most significant field is the first 2-tuple.

pref(*name*, *default=None*)
Return the value for the specified preference or the value specified as `default` if the preference is not set.

read_backup(*book_id*)
Return the OPF metadata backup for the book as a bytestring or None if no such backup exists.

remove_books(*book_ids*, *permanent=False*)
Remove the books specified by the book_ids from the database and delete their format files. If `permanent` is False, then the format files are not deleted.

remove_formats(*formats_map*, *db_only=False*)
Remove the specified formats from the specified books.

Parametry

- **formats_map** – A mapping of book_id to a list of formats to be removed from the book.
- **db_only** – If True, only remove the record for the format from the db, do not delete the actual format file from the filesystem.

remove_items(*field*, *item_ids*, *restrict_to_book_ids=None*)
Delete all items in the specified field with the specified ids. Returns the set of affected book ids. `restrict_to_book_ids` is an optional set of books ids. If specified the items will only be removed from those books.

rename_items(*field*, *item_id_to_new_name_map*, *change_index=True*, *restrict_to_book_ids=None*)
Rename items from a many-one or many-many field such as tags or series.

Parametry

- **change_index** – When renaming in a series-like field also change the series_index values.
- **restrict_to_book_ids** – An optional set of book ids for which the rename is to be performed, defaults to all books.

restore_book(*book_id*, *mi*, *last_modified*, *path*, *formats*)
Restore the book entry in the database for a book that already exists on the filesystem

restore_original_format(*book_id*, *original_fmt*)
Restore the specified format from the previously saved ORIGINAL_FORMAT, if any. Return True on success. The ORIGINAL_FORMAT is deleted after a successful restore.

safe_read_lock
A safe read lock is a lock that does nothing if the thread already has a write lock, otherwise it acquires a read lock. This is necessary to prevent DowngradeLockErrors, which can happen when updating the search cache in the presence of composite columns. Updating the search cache holds an exclusive lock, but

searching a composite column involves reading field values via ProxyMetadata which tries to get a shared lock. There may be other scenarios that trigger this as well.

This property returns a new lock object on every access. This lock object is not recursive (for performance) and must only be used in a with statement as `with cache.safe_read_lock:` otherwise bad things will happen.

save_original_format (*book_id*, *fmt*)
Save a copy of the specified format as ORIGINAL_FORMAT, overwriting any existing ORIGINAL_FORMAT.

search (*query*, *restriction=u''*, *virtual_fields=None*, *book_ids=None*)
Search the database for the specified query, returning a set of matched book ids.

Parametry

- **restriction** – A restriction that is ANDed to the specified query. Note that restrictions are cached, therefore the search for a AND b will be slower than a with restriction b.
- **virtual_fields** – Used internally (virtual fields such as on_device to search over).
- **book_ids** – If not None, a set of book ids for which books will be searched instead of searching all books.

set_conversion_options (*options*, *fmt=u'PIPE'*)
options must be a map of the form {book_id:conversion_options}

set_cover (*book_id_data_map*)
Set the cover for this book. data can be either a QImage, QPixmap, file object or bytestring. It can also be None, in which case any existing cover is removed.

set_field (*name*, *book_id_to_val_map*, *allow_case_change=True*, *do_path_update=True*)
Set the values of the field specified by `name`. Returns the set of all book ids that were affected by the change.

Parametry

- **book_id_to_val_map** – Mapping of book_ids to values that should be applied.
- **allow_case_change** – If True, the case of many-one or many-many fields will be changed. For example, if a book has the tag `tag1` and you set the tag for another book to `Tag1` then the both books will have the tag `Tag1` if allow_case_change is True, otherwise they will both have the tag `tag1`.
- **do_path_update** – Used internally, you should never change it.

set_metadata (*book_id*, *mi*, *ignore_errors=False*, *force_changes=False*, *set_title=True*, *set_authors=True*, *allow_case_change=False*)
Set metadata for the book *id* from the *Metadata* object *mi*

Setting force_changes=True will force set_metadata to update fields even if mi contains empty values. In this case, 'None' is distinguished from 'empty'. If mi.XXX is None, the XXX is not replaced, otherwise it is. The tags, identifiers, and cover attributes are special cases. Tags and identifiers cannot be set to None so then will always be replaced if force_changes is true. You must ensure that mi contains the values you want the book to have. Covers are always changed if a new cover is provided, but are never deleted. Also note that force_changes has no effect on setting title or authors.

set_pref (*name*, *val*)
Set the specified preference to the specified value. See also *pref()* (stránka 271).

tags_older_than (*tag*, *delta=None*, *must_have_tag=None*, *must_have_authors=None*)
Return the ids of all books having the tag `tag` that are older than the specified time. tag comparison is case insensitive.

Parametry

- **delta** – A timedelta object or None. If None, then all ids with the tag are returned.
- **must_have_tag** – If not None the list of matches will be restricted to books that have this tag
- **must_have_authors** – A list of authors. If not None the list of matches will be restricted to books that have these authors (case insensitive).

user_categories_for_books (*book_ids*, *proxy_metadata_map=None*)

Return the user categories for the specified books. proxy_metadata_map is optional and is useful for a performance boost, in contexts where a ProxyMetadata object for the books already exists. It should be a mapping of book_ids to their corresponding ProxyMetadata objects.

API Documentation for the ebook editing tools

The ebook editing tools consist of a *calibre.ebooks.oeb.polish.container.Container* (stránka 273) object that represents a book as a collection of HTML + resource files, and various tools that can be used to perform operations on the container. All the tools are in the form of module level functions in the various `calibre.ebooks.oeb.polish.*` modules. You obtain a container object for a book at a path like this:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

If you are writing a plugin for the ebook editor, you get the current container for the book being edited like this:

```
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

The Container object

class `calibre.ebooks.oeb.polish.container.`**Container** (*rootpath*, *opfpath*, *log*, *clone_data=None*)

A container represents an Open EBook as a directory full of files and an opf file. There are two important concepts:

- The root directory. This is the base of the ebook. All the ebooks files are inside this directory or in its sub-directories.
- Names: These are paths to the books' files relative to the root directory. They always contain POSIX separators and are unquoted. They can be thought of as canonical identifiers for files in the book. Most methods on the container object work with names. Names are always in the NFC unicode normal form.
- Clones: the container object supports efficient on-disk cloning, which is used to implement checkpoints in the ebook editor. In order to make this work, you should never access files on the filesystem directly. Instead, use *raw_data()* (stránka 275) or *open()* (stránka 275) to read/write to component files in the book.

When converting between hrefs and names use the methods provided by this class, they assume all hrefs are quoted.

abspath_to_name (*fullpath*, *root=None*)

Convert an absolute path to a canonical name relative to `root`

Parametry **root** – The base directory. By default the root for this container object is used.

add_file(*name*, *data*, *media_type=None*, *spine_index=None*)
Add a file to this container. Entries for the file are automatically created in the OPF manifest and spine (if the file is a text document)

add_name_to_manifest(*name*)
Add an entry to the manifest for a file with the specified name. Returns the manifest id.

book_type = u'oeb'
The type of book (epub for EPUB files and azw3 for AZW3 files)

commit(*outpath=None*, *keep_parsed=False*)
Commit all dirtied parsed objects to the filesystem and write out the ebook file at outpath. :param output: The path to write the saved ebook file to. If None, the path of the original book file is used. :param keep_parsed: If True the parsed representations of committed items are kept in the cache.

commit_item(*name*, *keep_parsed=False*)
Commit a parsed object to disk (it is serialized and written to the underlying file). If `keep_parsed` is True the parsed representation is retained in the cache. See also: *parsed()* (stránka 275)

decode(*data*, *normalize_to_nfc=True*)
Automatically decode `data` into a `unicode` object.

Parametry **normalize_to_nfc** – Normalize returned unicode to the NFC normal form as is required by both the EPUB and AZW3 formats.

dirty(*name*)
Mark the parsed object corresponding to name as dirty. See also: *parsed()* (stránka 275).

exists(*name*)
True iff a file corresponding to the canonical name exists. Note that this function suffers from the limitations of the underlying OS filesystem, in particular case (in)sensitivity. So on a case insensitive filesystem this will return True even if the case of name is different from the case of the underlying filesystem file. See also *has_name()* (stránka 274)

filesize(*name*)
Return the size in bytes of the file represented by the specified canonical name. Automatically handles dirtied parsed objects. See also: *parsed()* (stránka 275)

generate_item(*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)
Add an item to the manifest with href derived from the given name. Ensures uniqueness of href and id automatically. Returns generated item.

get_file_path_for_processing(*name*, *allow_modification=True*)
Similar to open() except that it returns a file path, instead of an open file object.

guess_type(*name*)
Return the expected mimetype for the specified file name based on its extension.

guide_type_map
Mapping of guide type to canonical name

has_name(*name*)
Return True iff a file with the same canonical name as that specified exists. Unlike *exists()* (stránka 274) this method is always case-sensitive.

href_to_name(*href*, *base=None*)
Convert an href (relative to base) to a name. base must be a name or None, in which case self.root is used.

insert_into_xml(*parent*, *item*, *index=None*)
Insert item into parent (or append if index is None), fixing indentation. Only works with self closing items.

is_dir = False
If this container represents an unzipped book (a directory)

iterlinks(*name*, *get_line_numbers=True*)
Iterate over all links in name. If get_line_numbers is True the yields results of the form (link, line_number, offset). Where line_number is the line_number at which the link occurs and offset is the number of characters from the start of the line. Note that offset could actually encompass several lines if not zero.

manifest_has_name(*name*)
Return True if the manifest has an entry corresponding to name

manifest_id_map
Mapping of manifest id to canonical names

manifest_type_map
Mapping of manifest media-type to list of canonical names of that media-type

mi
The metadata of this book as a Metadata object. Note that this object is constructed on the fly every time this property is requested, so use it sparingly.

name_to_abspath(*name*)
Convert a canonical name to an absolute OS dependant path

name_to_href(*name*, *base=None*)
Convert a name to a href relative to base, which must be a name or None in which case self.root is used as the base

names_that_must_not_be_changed
Set of names that must never be renamed. Depends on the ebook file format.

names_that_must_not_be_removed
Set of names that must never be deleted from the container. Depends on the ebook file format.

names_that_need_not_be_manifested
Set of names that are allowed to be missing from the manifest. Depends on the ebook file format.

open(*name*, *mode=u'rb'*)
Open the file pointed to by name for direct read/write. Note that this will commit the file if it is dirtied and remove it from the parse cache. You must finish with this file before accessing the parsed version of it again, or bad things will happen.

opf
The parsed OPF file

opf_get_or_create(*name*)
Convenience method to either return the first XML element with the specified name or create it under the opf:package element and then return it, if it does not already exist.

opf_version
The version set on the OPF's <package> element

opf_xpath(*expr*)
Convenience method to evaluate an XPath expression on the OPF file, has the opf: and dc: namespace prefixes pre-defined.

parsed(*name*)
Return a parsed representation of the file specified by name. For HTML and XML files an lxml tree is returned. For CSS files a cssutils stylesheet is returned. Note that parsed objects are cached for performance. If you make any changes to the parsed object, you must call *dirty()* (stránka 274) so that the container knows to update the cache. See also *replace()* (stránka 276).

raw_data(*name*, *decode=True*, *normalize_to_nfc=True*)
Return the raw data corresponding to the file specified by name

Parametry

- **decode** – If True and the file has a text based mimetype, decode it and return a unicode object instead of raw bytes.
- **normalize_to_nfc** – If True the returned unicode object is normalized to the NFC normal form as is required for the EPUB and AZW3 file formats.

relpath(*path*, *base=None*)
Convert an absolute path (with os separators) to a path relative to base (defaults to self.root). The relative path is *not* a name. Use *abspath_to_name()* (stránka 273) for that.

remove_from_spine(*spine_items*, *remove_if_no_longer_in_spine=True*)
Remove the specified items (by canonical name) from the spine. If `remove_if_no_longer_in_spine` is True, the items are also deleted from the book, not just from the spine.

remove_from_xml(*item*)
Removes item from parent, fixing indentation (works only with self closing items)

remove_item(*name*, *remove_from_guide=True*)
Remove the item identified by name from this container. This removes all references to the item in the OPF manifest, guide and spine as well as from any internal caches.

rename(*current_name*, *new_name*)
Renames a file from current_name to new_name. It automatically rebases all links inside the file if the directory the file is in changes. Note however, that links are not updated in the other files that could reference this file. This is for performance, such updates should be done once, in bulk.

replace(*name*, *obj*)
Replace the parsed object corresponding to name with obj, which must be a similar object, i.e. an lxml tree for HTML/XML or a cssutils stylesheet for a CSS file.

replace_links(*name*, *replace_func*)
Replace all links in name using replace_func, which must be a callable that accepts a URL and returns the replaced URL. It must also have a 'replaced' attribute that is set to True if any actual replacement is done. Convenient ways of creating such callables are using the `LinkReplacer` and `LinkRebaser` classes.

serialize_item(*name*)
Convert a parsed object (identified by canonical name) into a bytestring. See *parsed()* (stránka 275).

set_spine(*spine_items*)
Set the spine to be spine_items where spine_items is an iterable of the form (name, linear). Will raise an error if one of the names is not present in the manifest.

spine_items
An iterator yielding canonical name for every item in the books' spine. See also: *spine_iter* (stránka 276) and *spine_items* (stránka 276).

spine_iter
An iterator that yields item, name is_linear for every item in the books' spine. item is the lxml element, name is the canonical file name and is_linear is True if the item is linear. See also: *spine_names* (stránka 276) and *spine_items* (stránka 276).

spine_names
An iterator yielding name and is_linear for every item in the books' spine. See also: *spine_iter* (stránka 276) and *spine_items* (stránka 276).

Managing component files in a container

calibre.ebooks.oeb.polish.replace.**replace_links**(*container*, *link_map*, *frag_map=<function <lambda>>*, *replace_in_opf=False*)

Replace links to files in the container. Will iterate over all files in the container and change the specified links in them.

Parametry

- **link_map** – A mapping of old canonical name to new canonical name. For example: `{'images/old.png': 'images/new.png'}`
- **frag_map** – A callable that takes two arguments `(name, anchor)` and returns a new anchor. This is useful if you need to change the anchors in HTML files. By default, it does nothing.
- **replace_in_opf** – If False, links are not replaced in the OPF file.

calibre.ebooks.oeb.polish.replace.**rename_files**(*container*, *file_map*)

Rename files in the container, automatically updating all links to them.

Parametry **file_map** – A mapping of old canonical name to new canonical name, for example: `{'text/chapter1.html': 'chapter1.html'}`.

calibre.ebooks.oeb.polish.replace.**get_recommended_folders**(*container*, *names*)

Return the folders that are recommended for the given filenames. The recommendation is based on where the majority of files of the same type are located in the container. If no files of a particular type are present, the recommended folder is assumed to be the folder containing the OPF file.

Pretty printing and auto fixing parse errors

calibre.ebooks.oeb.polish.pretty.**fix_html**(*container*, *raw*)

Fix any parsing errors in the HTML represented as a string in raw. Fixing is done using the HTML5 parsing algorithm.

calibre.ebooks.oeb.polish.pretty.**fix_all_html**(*container*)

Fix any parsing errors in all HTML files in the container. Fixing is done using the HTML5 parsing algorithm.

calibre.ebooks.oeb.polish.pretty.**pretty_html**(*container*, *name*, *raw*)

Pretty print the HTML represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_css**(*container*, *name*, *raw*)

Pretty print the CSS represented as a string in raw

calibre.ebooks.oeb.polish.pretty.**pretty_xml**(*container*, *name*, *raw*)

Pretty print the XML represented as a string in raw. If `name` is the name of the OPF, extra OPF-specific prettying is performed.

calibre.ebooks.oeb.polish.pretty.**pretty_all**(*container*)

Pretty print all HTML/CSS/XML files in the container

Managing book jackets

calibre.ebooks.oeb.polish.jacket.**remove_jacket**(*container*)

Remove an existing jacket, if any. Returns False if no existing jacket was found.

calibre.ebooks.oeb.polish.jacket.**add_or_replace_jacket**(*container*)
Either create a new jacket from the book's metadata or replace an existing jacket. Returns True if an existing jacket was replaced.

Splitting and merging of files

calibre.ebooks.oeb.polish.split.**split**(*container*, *name*, *loc_or_xpath*, *before=True*, *totals=None*)
Split the file specified by name at the position specified by loc_or_xpath. Splitting automatically migrates all links and references to the affected files.

Parametry

- **loc_or_xpath** – Should be an XPath expression such as //h:div[@id="split_here"]. Can also be a *loc* which is used internally to implement splitting in the preview panel.
- **before** – If True the split occurs before the identified element otherwise after it.
- **totals** – Used internally

calibre.ebooks.oeb.polish.split.**multisplit**(*container*, *name*, *xpath*, *before=True*)
Split the specified file at multiple locations (all tags that match the specified XPath expression. See also: *split()* (stránka 278). Splitting automatically migrates all links and references to the affected files.

Parametry **before** – If True the splits occur before the identified element otherwise after it.

calibre.ebooks.oeb.polish.split.**merge**(*container*, *category*, *names*, *master*)
Merge the specified files into a single file, automatically migrating all links and references to the affected files. The file must all either be HTML or CSS files.

Parametry

- **category** – Must be either `'text'` for HTML files or `'styles'` for CSS files
- **names** – The list of files to be merged
- **master** – Which of the merged files is the *master* file, that is, the file that will remain after merging.

Managing covers

calibre.ebooks.oeb.polish.cover.**set_cover**(*container*, *cover_path*, *report=None*, *options=None*)
Set the cover of the book to the image pointed to by cover_path.

Parametry

- **cover_path** – Either the absolute path to an image file or the canonical name of an image in the book. When using an image in the book, you must also set options, see below.
- **report** – An optional callable that takes a single argument. It will be called with information about the tasks being processed.
- **options** – None or a dictionary that controls how the cover is set. The dictionary can have entries: **keep_aspect**: True or False (Preserve aspect ratio of covers in EPUB) **no_svg**: True or False (Use an SVG cover wrapper in the EPUB titlepage) **existing**: True or False (`cover_path` refers to an existing image in the book)

calibre.ebooks.oeb.polish.cover.**mark_as_cover**(*container*, *name*)
Mark the specified image as the cover image.

calibre.ebooks.oeb.polish.cover.**mark_as_titlepage**(*container*, *name*, *move_to_start=True*)
Mark the specified HTML file as the titlepage of the EPUB.

Parametry **move_to_start** – If True the HTML file is moved to the start of the spine

Working with CSS

calibre.ebooks.oeb.polish.fonts.**change_font**(*container*, *old_name*, *new_name=None*)
Change a font family from old_name to new_name. Changes all occurrences of the font family in stylesheets, style tags and style attributes. If the old_name refers to an embedded font, it is removed. You can set new_name to None to remove the font family instead of changing it.

calibre.ebooks.oeb.polish.css.**remove_unused_css**(*container*, *report=None*, *remove_unused_classes=False*)
Remove all unused CSS rules from the book. An unused CSS rule is one that does not match any actual content.

Parametry

- **report** – An optional callable that takes a single argument. It is called with information about the operations being performed.
- **remove_unused_classes** – If True, class attributes in the HTML that do not match any CSS rules are also removed.

calibre.ebooks.oeb.polish.css.**filter_css**(*container*, *properties*, *names=()*)
Remove the specified CSS properties from all CSS rules in the book.

Parametry

- **properties** – Set of properties to remove. For example: `{'font-family', 'color'}`.
- **names** – The files from which to remove the properties. Defaults to all HTML and CSS files in the book.

Working with the Table of Contents

calibre.ebooks.oeb.polish.toc.**from_xpaths**(*container*, *xpaths*)
Generate a Table of Contents from a list of XPath expressions. Each expression in the list corresponds to a level of the generate ToC. For example: `['//h:h1', '//h:h2', '//h:h3']` will generate a three level table of contents from the <h1>, <h2> and <h3> tags.

calibre.ebooks.oeb.polish.toc.**from_links**(*container*)
Generate a Table of Contents from links in the book.

calibre.ebooks.oeb.polish.toc.**from_files**(*container*)
Generate a Table of Contents from files in the book.

calibre.ebooks.oeb.polish.toc.**create_inline_toc**(*container*, *title=None*)
Create an inline (HTML) Table of Contents from an existing NCX table of contents.

Parametry **title** – The title for this table of contents.

Nástroj na úpravu knihy

class calibre.gui2.tweak_book.plugin.**Tool**
Nadtřídy: object The base class for individual tools in an Edit Book plugin. Useful members include:

- `self.plugin`: A reference to the *calibre.customize.Plugin* (stránka 194) object to which this tool belongs.
- self. *boss* (stránka 280)
- self. *gui* (stránka 280)

Methods that must be overridden in sub classes:

- *create_action()* (stránka 280)
- *register_shortcut()* (stránka 280)

name = None
Set this to a unique name it will be used as a key

allowed_in_toolbar = True
If True the user can choose to place this tool in the plugins toolbar

allowed_in_menu = True
If True the user can choose to place this tool in the plugins menu

toolbar_button_popup_mode = u'delayed'
The popup mode for the menu (if any) of the toolbar button. Possible values are 'delayed', 'instant', 'button'

boss
The *calibre.gui2.tweak_book.boss.Boss* (stránka 281) object. Used to control the user interface.

gui
The main window of the user interface

current_container
Return the current *calibre.ebooks.oeb.polish.container.Container* (stránka 273) object that represents the book being edited.

register_shortcut(*qaction*, *unique_name*, *default_keys=()*, *short_text=None*, *description=None*, ***extra_data*)
Register a keyboard shortcut that will trigger the specified `qaction`. This keyboard shortcut will become automatically customizable by the user in the Keyboard section of the editor preferences.

Parametry

- **qaction** – A QAction object, it will be triggered when the configured key combination is pressed by the user.
- **unique_name** – A unique name for this shortcut/action. It will be used internally, it must not be shared by any other actions in this plugin.
- **default_keys** – A list of the default keyboard shortcuts. If not specified no default shortcuts will be set. If the shortcuts specified here conflict with either builtin shortcuts or shortcuts from user configuration/other plugins, they will be ignored. In that case, users will have to configure the shortcuts manually via Preferences. For example: `default_keys=('Ctrl+J', 'F9')`.
- **short_text** – An optional short description of this action. If not specified the text from the QAction will be used.
- **description** – An optional longer description of this action, it will be used in the preferences entry for this shortcut.

create_action(*for_toolbar=True*)
Create a QAction that will be added to either the plugins toolbar or the plugins menu depending on `for_toolbar`. For example:

```
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
        # We want the toolbar button to have a popup menu
        menu = QMenu()
        ac.setMenu(menu)
        menu.addAction('Do something else')
        subaction = menu.addAction('And another')

        # Register a keyboard shortcut for this toolbar action be
        # careful to do this for only one of the toolbar action or
        # the menu action, not both.
        self.register_shortcut(ac, 'some-unique-name', default_keys=('Ctrl+K',))
    return ac
```

Viz také:

Method *register_shortcut()* (stránka 280).

Controlling the editor's user interface

The ebook editor's user interface is controlled by a single global *Boss* object. This has many useful methods that can be used in plugin code to perform common tasks.

class `calibre.gui2.tweak_book.boss.`**Boss**(*parent*, *notify=None*)

add_savepoint(*msg*)
Create a restore checkpoint with the name specified as `msg`

apply_container_update_to_gui(*mark_as_modified=True*)
Update all the components of the user interface to reflect the latest data in the current book container.

Parametry **mark_as_modified** – If True, the book will be marked as modified, so the user will be prompted to save it when quitting.

close_editor(*name*)
Close the editor that is editing the file specified by `name`

commit_all_editors_to_container()
Commit any changes that the user has made to files open in editors to the container. You should call this method before performing any actions on the current container

currently_editing
Return the name of the file being edited currently or None if no file is being edited

edit_file(*name*, *syntax=None*, *use_template=None*)
Open the file specified by name in an editor

Parametry

- **syntax** – The media type of the file, for example, `'text/html'`. If not specified it is guessed from the file extension.
- **use_template** – A template to initialize the opened editor with

open_book(*path=None*, *edit_file=None*, *clear_notify_data=True*, *open_folder=False*)
Open the ebook at `path` for editing. Will show an error if the ebook is not in a supported format or the current book has unsaved changes.

Parametry **edit_file** – The name of a file inside the newly opened book to start editing. Can also be a list of names.

rewind_savepoint()
Undo the previous creation of a restore checkpoint, useful if you create a checkpoint, then abort the operation with no changes

save_book()
Save the book. Saving is performed in the background

set_modified()
Mark the book as having been modified

show_current_diff(*allow_revert=True*, *to_container=None*)
Show the changes to the book from its last checkpointed state

Parametry

- **allow_revert** – If True the diff dialog will have a button to allow the user to revert all changes
- **to_container** – A container object to compare the current container to. If None, the previously checkpointed container is used

show_editor(*name*)
Show the editor that is editing the file specified by `name`

sync_preview_to_editor()
Sync the position of the preview panel to the current cursor position in the current editor

1.13 Slovník pojmů

RSS **RSS** *(Really Simple Syndication)* je formát webového informačního kanálu, který se používá k publikování často aktualizovaného obsahu, jako jsou novinové články, příspěvky blogu atd. Je to formát, který je obzvlášť vhodný pro čtení počítači, a je proto upřednostňovaným způsob získávání obsahu z webu do e-knihy. Existuje mnoho dalších formátů informačních kanálů používaných na internetu a calibre rozumí většině z nich. Zejména má dobrou podporu pro formát *ATOM*, který se obvykle používá pro blogy.

předpis Předpis je sada instrukcí, která učí calibre, jak převést na e-knihu online zdroje zpráv, jako jsou časopisy nebo blogy. Předpis je v podstatě kód Pythonu[111]. Jako takový je schopný převést na e-knihy libovolně složitý zdroj zpráv. Na nejjednodušší úrovni je to pouze sada proměnných, jako jsou URL, které dávají calibre dostatek informací pro přejití na internet a stažení zpráv.

HTML **HTML** *(značkovací jazyk pro hypertext = HyperText Markup Language)*, podmnožina SGML (univerzální značkovací metajazyk = Standard Generalized Markup Language) pro elektronické publikování, je konkrétní standard použití pro web.

CSS **CSS** *(šablony stylů = Cascading Style Sheets)* je jazyk použitý pro popsání, jak by měl být vykreslen dokument *HTML* (vizuální stylování).

API **API** *(rozhraní pro programování aplikací = Application Programming Interface)* je rozhraní zdrojového kódu, které knihovna poskytuje pro podporu požadavků na služby, které na něj mají být provedeny počítačovými programy.

[111] https://www.python.org

LRF **LRF** Formát e-knihy, který je čten čtečkami e-knih SONY.

URL **URL** *(jednotná adresa zdroje = Uniform Resource Locator)* například: `http://example.com`

regexp **Regulární výrazy** poskytují stručný a pružný způsob pro identifikaci řetězců zkoumaného textu, jako jsou určité znaky, slova nebo vzory znaků. Podívejte se na syntaxi regexp <https://docs.python.org/2.7/library/re.html>‘_ pro syntaxi regulárních výrazů používaných v Pythonu.

C

calibre.customize, 193
calibre.customize.conversion, 200
calibre.db.cache, 267
calibre.devices.interface, 202
calibre.ebooks.metadata.book.base, 155
calibre.ebooks.metadata.sources.base, 198
calibre.ebooks.oeb.polish.container, 273
calibre.ebooks.oeb.polish.cover, 278
calibre.ebooks.oeb.polish.css, 279
calibre.ebooks.oeb.polish.jacket, 277
calibre.ebooks.oeb.polish.pretty, 277
calibre.ebooks.oeb.polish.replace, 277
calibre.ebooks.oeb.polish.split, 278
calibre.ebooks.oeb.polish.toc, 279
calibre.gui2.tweak_book.boss, 281
calibre.gui2.tweak_book.plugin.Tool, 279
calibre.utils.formatter_functions, 146
calibre.web.feeds.news, 43

Symbols

–add-plugin, -a
calibre-customize volba příkazového řádku, 229
–add-simple-plugin
calibre-debug volba příkazového řádku, 230
–all
calibredb-backup_metadata volba příkazového řádku, 243
calibredb-export volba příkazového řádku, 238
–append, -a
calibredb-set_custom volba příkazového řádku, 241
–as-opf
calibredb-show_metadata volba příkazového řádku, 237
–ascending
calibredb-list volba příkazového řádku, 235
–asciiize
ebook-convert volba příkazového řádku, 247
–attachment, -a
calibre-smtp volba příkazového řádku, 233
–author-sort
ebook-convert volba příkazového řádku, 251
ebook-meta volba příkazového řádku, 253
–authors
ebook-convert volba příkazového řádku, 251
–authors, -a
calibredb-add volba příkazového řádku, 236
ebook-meta volba příkazového řádku, 253
fetch-ebook-metadata volba příkazového řádku, 257
–auto-reload
calibre-server volba příkazového řádku, 232
–base-dir, -d
web2disk volba příkazového řádku, 259
–base-font-size
ebook-convert volba příkazového řádku, 247
–book-producer
ebook-convert volba příkazového řádku, 251
–book-producer, -k
ebook-meta volba příkazového řádku, 253
–build-plugin, -b
calibre-customize volba příkazového řádku, 229
–categories, -r
calibredb-list_categories volba příkazového řádku, 242
–category
ebook-meta volba příkazového řádku, 253
–change-justification
ebook-convert volba příkazového řádku, 247
–chapter
ebook-convert volba příkazového řádku, 250
–chapter-mark
ebook-convert volba příkazového řádku, 250
–command, -c
calibre-debug volba příkazového řádku, 230
–comments
ebook-convert volba příkazového řádku, 252
–comments, -c
ebook-meta volba příkazového řádku, 253
–compress-images, -i
ebook-polish volba příkazového řádku, 255
–continue
ebook-viewer volba příkazového řádku, 256
–cover
ebook-convert volba příkazového řádku, 252
ebook-meta volba příkazového řádku, 253
–cover, -c
calibredb-add volba příkazového řádku, 236
ebook-polish volba příkazového řádku, 255
fetch-ebook-metadata volba příkazového řádku, 257
–csv, -c
calibredb-check_library volba příkazového řádku, 242
calibredb-list_categories volba příkazového řádku, 243
–customize-plugin
calibre-customize volba příkazového řádku, 230
–daemonize
calibre-server volba příkazového řádku, 232
–date, -d
ebook-meta volba příkazového řádku, 254
–debug-device-driver, -d

calibre-debug volba příkazového řádku, 230
–debug-javascript
ebook-viewer volba příkazového řádku, 256
–debug-pipeline, -d
ebook-convert volba příkazového řádku, 252
–default-programs
calibre-debug volba příkazového řádku, 230
–delay
web2disk volba příkazového řádku, 259
–detach
calibre volba příkazového řádku, 229
ebook-edit volba příkazového řádku, 253
ebook-viewer volba příkazového řádku, 256
–details, -d
calibredb-custom_columns volba příkazového řádku, 241
–develop
calibre-server volba příkazového řádku, 232
–diff
calibre-debug volba příkazového řádku, 230
–disable-dehyphenate
ebook-convert volba příkazového řádku, 249
–disable-delete-blank-paragraphs
ebook-convert volba příkazového řádku, 249
–disable-fix-indents
ebook-convert volba příkazového řádku, 249
–disable-font-rescaling
ebook-convert volba příkazového řádku, 247
–disable-format-scene-breaks
ebook-convert volba příkazového řádku, 249
–disable-hyphenation
lrfviewer volba příkazového řádku, 258
–disable-italicize-common-cases
ebook-convert volba příkazového řádku, 249
–disable-markup-chapter-headings
ebook-convert volba příkazového řádku, 249
–disable-plugin
calibre-customize volba příkazového řádku, 230
–disable-remove-fake-margins
ebook-convert volba příkazového řádku, 250
–disable-renumber-headings
ebook-convert volba příkazového řádku, 249
–disable-unwrap-lines
ebook-convert volba příkazového řádku, 249
–display
calibredb-add_custom_column volba příkazového řádku, 240
–dont-asciiize
calibredb-export volba příkazového řádku, 238
–dont-download-stylesheets
web2disk volba příkazového řádku, 259
–dont-notify-gui
volba příkazového řádku, 234
–dont-output-resources
lrf2lrs volba příkazového řádku, 257
–dont-replace
calibredb-add_format volba příkazového řádku, 237
–dont-save-cover
calibredb-export volba příkazového řádku, 238
–dont-split-on-page-breaks
ebook-convert volba příkazového řádku, 246
–dont-update-metadata
calibredb-export volba příkazového řádku, 238
–dont-write-opf
calibredb-export volba příkazového řádku, 238
–duplicate-links-in-toc
ebook-convert volba příkazového řádku, 251
–duplicates, -d
calibredb-add volba příkazového řádku, 236
–edit-book, -t
calibre-debug volba příkazového řádku, 231
–embed-all-fonts
ebook-convert volba příkazového řádku, 247
–embed-font-family
ebook-convert volba příkazového řádku, 247
–embed-fonts, -e
ebook-polish volba příkazového řádku, 255
–empty, -e
calibredb-add volba příkazového řádku, 236
–enable-heuristics
ebook-convert volba příkazového řádku, 249
–enable-plugin
calibre-customize volba příkazového řádku, 230
–encoding
web2disk volba příkazového řádku, 259
–encryption-method, -e
calibre-smtp volba příkazového řádku, 234
–epub-flatten
ebook-convert volba příkazového řádku, 246
–epub-inline-toc
ebook-convert volba příkazového řádku, 246
–epub-toc-at-end
ebook-convert volba příkazového řádku, 246
–exec-file, -e
calibre-debug volba příkazového řádku, 231
–expand-css
ebook-convert volba příkazového řádku, 247
–explode-book, -x
calibre-debug volba příkazového řádku, 231
–extra-css
ebook-convert volba příkazového řádku, 247
–extract-to
ebook-convert volba příkazového řádku, 246
–field, -f
calibredb-set_metadata volba příkazového řádku, 238
–fields
calibredb-catalog volba příkazového řádku, 239

–fields, -f
calibredb-list volba příkazového řádku, 235
–filter-css
ebook-convert volba příkazového řádku, 247
–filter-regexp
web2disk volba příkazového řádku, 259
–flow-size
ebook-convert volba příkazového řádku, 246
–font-size-mapping
ebook-convert volba příkazového řádku, 248
–for-machine
calibredb-list volba příkazového řádku, 235
–force, -f
calibredb-remove_custom_column volba příkazového řádku, 241
–fork, -f
calibre-smtp volba příkazového řádku, 233
–formats
calibredb-export volba příkazového řádku, 239
–from-opf
ebook-meta volba příkazového řádku, 254
–full-screen, -f
ebook-viewer volba příkazového řádku, 256
–get-cover
ebook-meta volba příkazového řádku, 254
–gui, -g
calibre-debug volba příkazového řádku, 231
–gui-debug
calibre-debug volba příkazového řádku, 231
–help, -h
calibre volba příkazového řádku, 229
calibre-customize volba příkazového řádku, 230
calibre-debug volba příkazového řádku, 231
calibre-server volba příkazového řádku, 232
calibre-smtp volba příkazového řádku, 233
calibredb-add volba příkazového řádku, 236
calibredb-add_custom_column volba příkazového řádku, 240
calibredb-add_format volba příkazového řádku, 237
calibredb-backup_metadata volba příkazového řádku, 243
calibredb-catalog volba příkazového řádku, 239
calibredb-check_library volba příkazového řádku, 242
calibredb-clone volba příkazového řádku, 243
calibredb-custom_columns volba příkazového řádku, 241
calibredb-embed_metadata volba příkazového řádku, 244
calibredb-export volba příkazového řádku, 239
calibredb-list volba příkazového řádku, 235
calibredb-list_categories volba příkazového řádku, 243
calibredb-remove volba příkazového řádku, 236
calibredb-remove_custom_column volba příkazového řádku, 241
calibredb-remove_format volba příkazového řádku, 237
calibredb-restore_database volba příkazového řádku, 242
calibredb-saved_searches volba příkazového řádku, 240
calibredb-search volba příkazového řádku, 244
calibredb-set_custom volba příkazového řádku, 241
calibredb-set_metadata volba příkazového řádku, 238
calibredb-show_metadata volba příkazového řádku, 237
ebook-convert volba příkazového řádku, 245
ebook-edit volba příkazového řádku, 253
ebook-meta volba příkazového řádku, 254
ebook-polish volba příkazového řádku, 255
ebook-viewer volba příkazového řádku, 256
fetch-ebook-metadata volba příkazového řádku, 257
lrf2lrs volba příkazového řádku, 257
lrfviewer volba příkazového řádku, 258
lrs2lrf volba příkazového řádku, 258
web2disk volba příkazového řádku, 259
–html-unwrap-factor
ebook-convert volba příkazového řádku, 249
–identifier
ebook-meta volba příkazového řádku, 254
–identifier, -I
calibredb-add volba příkazového řádku, 236
–ids, -i
calibredb-catalog volba příkazového řádku, 239
–ignore-plugins
calibre volba příkazového řádku, 229
–ignore_extensions, -e
calibredb-check_library volba příkazového řádku, 242
–ignore_names, -n
calibredb-check_library volba příkazového řádku, 242
–index, -i
ebook-meta volba příkazového řádku, 254
–input-encoding
ebook-convert volba příkazového řádku, 246
–input-profile
ebook-convert volba příkazového řádku, 245
–insert-blank-line
ebook-convert volba příkazového řádku, 248
–insert-blank-line-size
ebook-convert volba příkazového řádku, 248
–insert-metadata
ebook-convert volba příkazového řádku, 250
–inspect-mobi, -m
calibre-debug volba příkazového řádku, 231

–is-multiple
calibredb-add_custom_column volba příkazového řádku, 240
–isbn
ebook-convert volba příkazového řádku, 252
ebook-meta volba příkazového řádku, 254
–isbn, -i
calibredb-add volba příkazového řádku, 236
fetch-ebook-metadata volba příkazového řádku, 257
–item_count, -i
calibredb-list_categories volba příkazového řádku, 243
–jacket, -j
ebook-polish volba příkazového řádku, 255
–keep-ligatures
ebook-convert volba příkazového řádku, 248
–language
ebook-convert volba příkazového řádku, 252
–language, -l
ebook-meta volba příkazového řádku, 254
–languages, -l
calibredb-add volba příkazového řádku, 236
–level1-toc
ebook-convert volba příkazového řádku, 251
–level2-toc
ebook-convert volba příkazového řádku, 251
–level3-toc
ebook-convert volba příkazového řádku, 251
–library-path
volba příkazového řádku, 234
–limit
calibredb-list volba příkazového řádku, 235
–limit, -l
calibredb-search volba příkazového řádku, 244
–line-height
ebook-convert volba příkazového řádku, 248
–line-width, -w
calibredb-list volba příkazového řádku, 235
–linearize-tables
ebook-convert volba příkazového řádku, 248
–list-fields, -l
calibredb-set_metadata volba příkazového řádku, 238
–list-plugins, -l
calibre-customize volba příkazového řádku, 230
–list-recipes
ebook-convert volba příkazového řádku, 245
–localhost, -l
calibre-smtp volba příkazového řádku, 233
–lrf-bookid
ebook-meta volba příkazového řádku, 254
–lrs
lrs2lrf volba příkazového řádku, 258
–margin-bottom
ebook-convert volba příkazového řádku, 248
–margin-left
ebook-convert volba příkazového řádku, 248
–margin-right
ebook-convert volba příkazového řádku, 248
–margin-top
ebook-convert volba příkazového řádku, 248
–match-regexp
web2disk volba příkazového řádku, 259
–max-cover
calibre-server volba příkazového řádku, 232
–max-files, -n
web2disk volba příkazového řádku, 259
–max-opds-items
calibre-server volba příkazového řádku, 232
–max-opds-ungrouped-items
calibre-server volba příkazového řádku, 232
–max-recursions, -r
web2disk volba příkazového řádku, 259
–max-toc-links
ebook-convert volba příkazového řádku, 251
–minimum-line-height
ebook-convert volba příkazového řádku, 248
–new-server
calibre-debug volba příkazového řádku, 231
–no-chapters-in-toc
ebook-convert volba příkazového řádku, 251
–no-default-epub-cover
ebook-convert volba příkazového řádku, 246
–no-svg-cover
ebook-convert volba příkazového řádku, 246
–no-update-check
calibre volba příkazového řádku, 229
–one-book-per-directory, -1
calibredb-add volba příkazového řádku, 236
–only-formats, -f
calibredb-embed_metadata volba příkazového řádku, 244
–open-at
ebook-viewer volba příkazového řádku, 256
–opf, -o
ebook-polish volba příkazového řádku, 255
fetch-ebook-metadata volba příkazového řádku, 257
–outbox, -o
calibre-smtp volba příkazového řádku, 233
–output, -o
lrf2lrs volba příkazového řádku, 257
lrs2lrf volba příkazového řádku, 258
–output-profile
ebook-convert volba příkazového řádku, 245
–page-breaks-before
ebook-convert volba příkazového řádku, 250
–password
calibre-server volba příkazového řádku, 232

–password, -p
calibre-smtp volba příkazového řádku, 234
–paths
calibre-debug volba příkazového řádku, 231
–pidfile
calibre-server volba příkazového řádku, 232
–port
calibre-smtp volba příkazového řádku, 234
–port, -p
calibre-server volba příkazového řádku, 232
–prefer-metadata-cover
ebook-convert volba příkazového řádku, 250
–prefix
calibredb-list volba příkazového řádku, 235
–preserve-cover-aspect-ratio
ebook-convert volba příkazového řádku, 246
–pretty-print
ebook-convert volba příkazového řádku, 247
–profile
lrfviewer volba příkazového řádku, 258
–pubdate
ebook-convert volba příkazového řádku, 252
–publisher
ebook-convert volba příkazového řádku, 252
–publisher, -p
ebook-meta volba příkazového řádku, 254
–py-console, -p
calibre-debug volba příkazového řádku, 231
–quote, -q
calibredb-list_categories volba příkazového řádku, 243
–raise-window
ebook-viewer volba příkazového řádku, 256
–rating
ebook-convert volba příkazového řádku, 252
–rating, -r
ebook-meta volba příkazového řádku, 254
–read-metadata-from-opf, -m
ebook-convert volba příkazového řádku, 252
–really-do-it, -r
calibredb-restore_database volba příkazového řádku, 242
–recurse, -r
calibredb-add volba příkazového řádku, 236
–reinitialize-db
calibre-debug volba příkazového řádku, 231
–relay, -r
calibre-smtp volba příkazového řádku, 234
–remove-first-image
ebook-convert volba příkazového řádku, 250
–remove-jacket
ebook-polish volba příkazového řádku, 255
–remove-paragraph-spacing
ebook-convert volba příkazového řádku, 248
–remove-paragraph-spacing-indent-size
ebook-convert volba příkazového řádku, 248
–remove-plugin, -r
calibre-customize volba příkazového řádku, 230
–remove-unused-css, -u
ebook-polish volba příkazového řádku, 255
–replace-scene-breaks
ebook-convert volba příkazového řádku, 249
–replace-whitespace
calibredb-export volba příkazového řádku, 239
–report, -r
calibredb-check_library volba příkazového řádku, 242
–restriction
calibre-server volba příkazového řádku, 232
–run-plugin, -r
calibre-debug volba příkazového řádku, 231
–search, -s
calibredb-catalog volba příkazového řádku, 239
calibredb-list volba příkazového řádku, 235
–search-replace
ebook-convert volba příkazového řádku, 250
–separator
calibredb-list volba příkazového řádku, 235
–separator, -s
calibredb-list_categories volba příkazového řádku, 243
–series
ebook-convert volba příkazového řádku, 252
–series, -s
calibredb-add volba příkazového řádku, 236
ebook-meta volba příkazového řádku, 254
–series-index
ebook-convert volba příkazového řádku, 252
–series-index, -S
calibredb-add volba příkazového řádku, 236
–shutdown-running-calibre, -s
calibre volba příkazového řádku, 229
calibre-debug volba příkazového řádku, 231
–single-dir
calibredb-export volba příkazového řádku, 239
–smarten-punctuation
ebook-convert volba příkazového řádku, 249
–smarten-punctuation, -p
ebook-polish volba příkazového řádku, 255
–sort-by
calibredb-catalog volba příkazového řádku, 240
calibredb-list volba příkazového řádku, 235
–sr1-replace
ebook-convert volba příkazového řádku, 250
–sr1-search
ebook-convert volba příkazového řádku, 250
–sr2-replace
ebook-convert volba příkazového řádku, 250

–sr2-search
ebook-convert volba příkazového řádku, 250
–sr3-replace
ebook-convert volba příkazového řádku, 250
–sr3-search
ebook-convert volba příkazového řádku, 250
–start-in-tray
calibre volba příkazového řádku, 229
–start-reading-at
ebook-convert volba příkazového řádku, 251
–subject, -s
calibre-smtp volba příkazového řádku, 233
–subset-embedded-fonts
ebook-convert volba příkazového řádku, 249
–subset-font, -f
calibre-debug volba příkazového řádku, 231
–subset-fonts, -f
ebook-polish volba příkazového řádku, 255
–tags
ebook-convert volba příkazového řádku, 252
ebook-meta volba příkazového řádku, 254
–tags, -T
calibredb-add volba příkazového řádku, 236
–template
calibredb-export volba příkazového řádku, 239
–test-build
calibre-debug volba příkazového řádku, 231
–thread-pool
calibre-server volba příkazového řádku, 232
–timefmt
calibredb-export volba příkazového řádku, 239
–timeout, -d
fetch-ebook-metadata volba příkazového řádku, 257
–timeout, -t
calibre-server volba příkazového řádku, 232
calibre-smtp volba příkazového řádku, 233
web2disk volba příkazového řádku, 259
–timestamp
ebook-convert volba příkazového řádku, 252
–title
ebook-convert volba příkazového řádku, 252
–title, -t
calibredb-add volba příkazového řádku, 236
ebook-meta volba příkazového řádku, 254
fetch-ebook-metadata volba příkazového řádku, 257
–title-sort
ebook-convert volba příkazového řádku, 252
ebook-meta volba příkazového řádku, 254
–to-dir
calibredb-export volba příkazového řádku, 239
–to-lowercase
calibredb-export volba příkazového řádku, 239
–to-opf
ebook-meta volba příkazového řádku, 254
–toc-filter
ebook-convert volba příkazového řádku, 251
–toc-threshold
ebook-convert volba příkazového řádku, 251
–toc-title
ebook-convert volba příkazového řádku, 247
–unsmarten-punctuation
ebook-convert volba příkazového řádku, 249
–url-prefix
calibre-server volba příkazového řádku, 232
–use-auto-toc
ebook-convert volba příkazového řádku, 251
–username
calibre-server volba příkazového řádku, 233
–username, -u
calibre-smtp volba příkazového řádku, 234
–verbose
ebook-polish volba příkazového řádku, 256
lrf2lrs volba příkazového řádku, 257
lrfviewer volba příkazového řádku, 258
lrs2lrf volba příkazového řádku, 258
web2disk volba příkazového řádku, 259
–verbose, -v
calibre volba příkazového řádku, 229
calibre-smtp volba příkazového řádku, 233
calibredb-catalog volba příkazového řádku, 240
ebook-convert volba příkazového řádku, 252
fetch-ebook-metadata volba příkazového řádku, 257
–version
calibre volba příkazového řádku, 229
calibre-customize volba příkazového řádku, 230
calibre-debug volba příkazového řádku, 231
calibre-server volba příkazového řádku, 233
calibre-smtp volba příkazového řádku, 233
calibredb-add volba příkazového řádku, 236
calibredb-add_custom_column volba příkazového řádku, 240
calibredb-add_format volba příkazového řádku, 237
calibredb-backup_metadata volba příkazového řádku, 243
calibredb-catalog volba příkazového řádku, 240
calibredb-check_library volba příkazového řádku, 242
calibredb-clone volba příkazového řádku, 244
calibredb-custom_columns volba příkazového řádku, 241
calibredb-embed_metadata volba příkazového řádku, 244
calibredb-export volba příkazového řádku, 239
calibredb-list volba příkazového řádku, 235
calibredb-list_categories volba příkazového řádku, 243
calibredb-remove volba příkazového řádku, 237

calibredb-remove_custom_column volba příkazového řádku, 241
calibredb-remove_format volba příkazového řádku, 237
calibredb-restore_database volba příkazového řádku, 242
calibredb-saved_searches volba příkazového řádku, 240
calibredb-search volba příkazového řádku, 244
calibredb-set_custom volba příkazového řádku, 241
calibredb-set_metadata volba příkazového řádku, 238
calibredb-show_metadata volba příkazového řádku, 237
ebook-convert volba příkazového řádku, 246
ebook-edit volba příkazového řádku, 253
ebook-meta volba příkazového řádku, 254
ebook-polish volba příkazového řádku, 256
ebook-viewer volba příkazového řádku, 256
fetch-ebook-metadata volba příkazového řádku, 257
lrf2lrs volba příkazového řádku, 257
lrfviewer volba příkazového řádku, 258
lrs2lrf volba příkazového řádku, 258
web2disk volba příkazového řádku, 259
–viewer, -w
calibre-debug volba příkazového řádku, 231
–visual-debug
lrfviewer volba příkazového řádku, 258
–white-background
lrfviewer volba příkazového řádku, 258
–width, -w
calibredb-list_categories volba příkazového řádku, 243
–with-library
calibre volba příkazového řádku, 229
calibre-server volba příkazového řádku, 233

A

abort_article() (metoda calibre.web.feeds.news.BasicNewsRecipe), 43
abort_recipe_processing() (metoda calibre.web.feeds.news.BasicNewsRecipe), 43
abspath_to_name() (metoda calibre.ebooks.oeb.polish.container.Container), 273
accept_drag_move_event() (metoda calibre.gui2.actions.InterfaceAction), 213
accept_enter_event() (metoda calibre.gui2.actions.InterfaceAction), 213
accepts_drops (atribut calibre.gui2.actions.InterfaceAction), 213
action_add_menu (atribut calibre.gui2.actions.InterfaceAction), 212
action_menu_clone_qaction (atribut calibre.gui2.actions.InterfaceAction), 213
action_spec (atribut calibre.gui2.actions.InterfaceAction), 212
action_type (atribut calibre.gui2.actions.InterfaceAction), 213
add_annotation_to_library() (metoda calibre.devices.usbms.device.Device), 211
add_book() (metoda calibre.devices.interface.BookList), 210
add_books() (metoda calibre.db.cache.Cache), 267
add_books_to_metadata() (třídní metoda calibre.devices.interface.DevicePlugin), 207
add_custom_book_data() (metoda calibre.db.cache.Cache), 267
add_file() (metoda calibre.ebooks.oeb.polish.container.Container), 273
add_format() (metoda calibre.db.cache.Cache), 267
add_name_to_manifest() (metoda calibre.ebooks.oeb.polish.container.Container), 274
add_or_replace_jacket() (v modulu calibre.ebooks.oeb.polish.jacket), 277
add_savepoint() (metoda calibre.gui2.tweak_book.boss.Boss), 281
add_toc_thumbnail() (metoda calibre.web.feeds.news.BasicNewsRecipe), 43
adeify_images() (třídní metoda calibre.web.feeds.news.BasicNewsRecipe), 43
all_book_ids() (metoda calibre.db.cache.Cache), 268
all_field_for() (metoda calibre.db.cache.Cache), 268
all_field_ids() (metoda calibre.db.cache.Cache), 268
all_field_keys() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
all_field_names() (metoda calibre.db.cache.Cache), 268
all_non_none_fields() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
allowed_in_menu (atribut calibre.gui2.tweak_book.plugin.Tool), 280
allowed_in_toolbar (atribut calibre.gui2.tweak_book.plugin.Tool), 280
API, **282**
apply_container_update_to_gui() (metoda calibre.gui2.tweak_book.boss.Boss), 281
articles_are_obfuscated (atribut calibre.web.feeds.news.BasicNewsRecipe), 47

ASK_TO_ALLOW_CONNECT (atribut calibre.devices.interface.DevicePlugin), 204
author (atribut calibre.customize.Plugin), 194
author_data() (metoda calibre.db.cache.Cache), 268
author_sort_from_authors() (metoda calibre.db.cache.Cache), 268
auto_cleanup (atribut calibre.web.feeds.news.BasicNewsRecipe), 47
auto_cleanup_keep (atribut calibre.web.feeds.news.BasicNewsRecipe), 47
auto_repeat (atribut calibre.gui2.actions.InterfaceAction), 212
auto_trim_covers (atribut calibre.ebooks.metadata.sources.base.Source), 198

B

BasicNewsRecipe (třída v calibre.web.feeds.news), 43
BCD (atribut calibre.devices.interface.DevicePlugin), 203
book_type (atribut calibre.ebooks.oeb.polish.container.Container), 274
BookList (třída v calibre.devices.interface), 209
books() (metoda calibre.devices.interface.DevicePlugin), 206
books_for_field() (metoda calibre.db.cache.Cache), 268
boss (atribut calibre.gui2.tweak_book.plugin.Tool), 280
Boss (třída v calibre.gui2.tweak_book.boss), 281
BuiltinAdd (třída v calibre.utils.formatter_functions), 146
BuiltinAnd (třída v calibre.utils.formatter_functions), 146
BuiltinApproximateFormats (třída v calibre.utils.formatter_functions), 147
BuiltinAssign (třída v calibre.utils.formatter_functions), 153
BuiltinAuthorLinks (třída v calibre.utils.formatter_functions), 147
BuiltinAuthorSorts (třída v calibre.utils.formatter_functions), 148
BuiltinBooksize (třída v calibre.utils.formatter_functions), 148
BuiltinCapitalize (třída v calibre.utils.formatter_functions), 154
BuiltinCmp (třída v calibre.utils.formatter_functions), 153
BuiltinContains (třída v calibre.utils.formatter_functions), 150
BuiltinCount (třída v calibre.utils.formatter_functions), 151
BuiltinCurrentLibraryName (třída v calibre.utils.formatter_functions), 148
BuiltinCurrentLibraryPath (třída v calibre.utils.formatter_functions), 148
BuiltinDaysBetween (třída v calibre.utils.formatter_functions), 146
BuiltinDivide (třída v calibre.utils.formatter_functions), 146
BuiltinEval (třída v calibre.utils.formatter_functions), 153
BuiltinField (třída v calibre.utils.formatter_functions), 148
BuiltinFinishFormatting (třída v calibre.utils.formatter_functions), 147
BuiltinFirstMatchingCmp (třída v calibre.utils.formatter_functions), 154
BuiltinFirstNonEmpty (třída v calibre.utils.formatter_functions), 150
BuiltinFormatDate (třída v calibre.utils.formatter_functions), 147
BuiltinFormatNumber (třída v calibre.utils.formatter_functions), 147
BuiltinFormatsModtimes (třída v calibre.utils.formatter_functions), 148
BuiltinFormatsPaths (třída v calibre.utils.formatter_functions), 148
BuiltinFormatsSizes (třída v calibre.utils.formatter_functions), 149
BuiltinHasCover (třída v calibre.utils.formatter_functions), 149
BuiltinHumanReadable (třída v calibre.utils.formatter_functions), 147
BuiltinIdentifierInList (třída v calibre.utils.formatter_functions), 151
BuiltinIfempty (třída v calibre.utils.formatter_functions), 150
BuiltinInList (třída v calibre.utils.formatter_functions), 151
BuiltinLanguageCodes (třída v calibre.utils.formatter_functions), 149
BuiltinLanguageStrings (třída v calibre.utils.formatter_functions), 149
BuiltinListDifference (třída v calibre.utils.formatter_functions), 151
BuiltinListEquals (třída v calibre.utils.formatter_functions), 152
BuiltinListIntersection (třída v calibre.utils.formatter_functions), 152
BuiltinListitem (třída v calibre.utils.formatter_functions), 151
BuiltinListRe (třída v calibre.utils.formatter_functions), 152
BuiltinListReGroup (třída v calibre.utils.formatter_functions), 152
BuiltinListSort (třída v calibre.utils.formatter_functions), 152
BuiltinListUnion (třída v calibre.utils.formatter_functions), 152
BuiltinLookup (třída v calibre.utils.formatter_functions),

150
BuiltinLowercase (třída v calibre.utils.formatter_functions), 154
BuiltinMultiply (třída v calibre.utils.formatter_functions), 146
BuiltinNot (třída v calibre.utils.formatter_functions), 146
BuiltinOndevice (třída v calibre.utils.formatter_functions), 149
BuiltinOr (třída v calibre.utils.formatter_functions), 146
BuiltinPrint (třída v calibre.utils.formatter_functions), 153
BuiltinRawField (třída v calibre.utils.formatter_functions), 149
BuiltinRawList (třída v calibre.utils.formatter_functions), 149
BuiltinRe (třída v calibre.utils.formatter_functions), 154
BuiltinReGroup (třída v calibre.utils.formatter_functions), 154
BuiltinSelect (třída v calibre.utils.formatter_functions), 151
BuiltinSeriesSort (třída v calibre.utils.formatter_functions), 149
BuiltinShorten (třída v calibre.utils.formatter_functions), 155
BuiltinStrcat (třída v calibre.utils.formatter_functions), 155
BuiltinStrcatMax (třída v calibre.utils.formatter_functions), 155
BuiltinStrcmp (třída v calibre.utils.formatter_functions), 154
BuiltinStrInList (třída v calibre.utils.formatter_functions), 151
BuiltinStrlen (třída v calibre.utils.formatter_functions), 155
BuiltinSubitems (třída v calibre.utils.formatter_functions), 152
BuiltinSublist (třída v calibre.utils.formatter_functions), 153
BuiltinSubstr (třída v calibre.utils.formatter_functions), 155
BuiltinSubtract (třída v calibre.utils.formatter_functions), 146
BuiltinSwapAroundComma (třída v calibre.utils.formatter_functions), 155
BuiltinSwitch (třída v calibre.utils.formatter_functions), 150
BuiltinTemplate (třída v calibre.utils.formatter_functions), 153
BuiltinTest (třída v calibre.utils.formatter_functions), 150
BuiltinTitlecase (třída v calibre.utils.formatter_functions), 154
BuiltinToday (třída v calibre.utils.formatter_functions), 146
BuiltinTransliterate (třída v calibre.utils.formatter_functions), 155
BuiltinUppercase (třída v calibre.utils.formatter_functions), 154
BuiltinUserCategories (třída v calibre.utils.formatter_functions), 149
BuiltinVirtualLibraries (třída v calibre.utils.formatter_functions), 150

C

Cache (třída v calibre.db.cache), 267
cached_cover_url_is_reliable (atribut calibre.ebooks.metadata.sources.base.Source), 198
calibre volba příkazového řádku
–detach, 229
–help, -h, 229
–ignore-plugins, 229
–no-update-check, 229
–shutdown-running-calibre, -s, 229
–start-in-tray, 229
–verbose, -v, 229
–version, 229
–with-library, 229
calibre-customize volba příkazového řádku
–add-plugin, -a, 229
–build-plugin, -b, 229
–customize-plugin, 230
–disable-plugin, 230
–enable-plugin, 230
–help, -h, 230
–list-plugins, -l, 230
–remove-plugin, -r, 230
–version, 230
calibre-debug volba příkazového řádku
–add-simple-plugin, 230
–command, -c, 230
–debug-device-driver, -d, 230
–default-programs, 230
–diff, 230
–edit-book, -t, 231
–exec-file, -e, 231
–explode-book, -x, 231
–gui, -g, 231
–gui-debug, 231
–help, -h, 231
–inspect-mobi, -m, 231
–new-server, 231
–paths, 231
–py-console, -p, 231
–reinitialize-db, 231
–run-plugin, -r, 231
–shutdown-running-calibre, -s, 231
–subset-font, -f, 231
–test-build, 231

–version, 231
–viewer, -w, 231
calibre-server volba příkazového řádku
–auto-reload, 232
–daemonize, 232
–develop, 232
–help, -h, 232
–max-cover, 232
–max-opds-items, 232
–max-opds-ungrouped-items, 232
–password, 232
–pidfile, 232
–port, -p, 232
–restriction, 232
–thread-pool, 232
–timeout, -t, 232
–url-prefix, 232
–username, 233
–version, 233
–with-library, 233
calibre-smtp volba příkazového řádku
–attachment, -a, 233
–encryption-method, -e, 234
–fork, -f, 233
–help, -h, 233
–localhost, -l, 233
–outbox, -o, 233
–password, -p, 234
–port, 234
–relay, -r, 234
–subject, -s, 233
–timeout, -t, 233
–username, -u, 234
–verbose, -v, 233
–version, 233
calibre.customize (modul), 193
calibre.customize.conversion (modul), 200
calibre.db.cache (modul), 267
calibre.devices.interface (modul), 202
calibre.ebooks.metadata.book.base (modul), 155
calibre.ebooks.metadata.sources.base (modul), 198
calibre.ebooks.oeb.polish.container (modul), 273
calibre.ebooks.oeb.polish.cover (modul), 278
calibre.ebooks.oeb.polish.css (modul), 279
calibre.ebooks.oeb.polish.jacket (modul), 277
calibre.ebooks.oeb.polish.pretty (modul), 277
calibre.ebooks.oeb.polish.replace (modul), 277
calibre.ebooks.oeb.polish.split (modul), 278
calibre.ebooks.oeb.polish.toc (modul), 279
calibre.gui2.tweak_book.boss (modul), 281
calibre.gui2.tweak_book.plugin.Tool (modul), 279
calibre.utils.formatter_functions (modul), 146
calibre.web.feeds.news (modul), 43
calibredb-add volba příkazového řádku
–authors, -a, 236
–cover, -c, 236
–duplicates, -d, 236
–empty, -e, 236
–help, -h, 236
–identifier, -I, 236
–isbn, -i, 236
–languages, -l, 236
–one-book-per-directory, -1, 236
–recurse, -r, 236
–series, -s, 236
–series-index, -S, 236
–tags, -T, 236
–title, -t, 236
–version, 236
calibredb-add_custom_column volba příkazového řádku
–display, 240
–help, -h, 240
–is-multiple, 240
–version, 240
calibredb-add_format volba příkazového řádku
–dont-replace, 237
–help, -h, 237
–version, 237
calibredb-backup_metadata volba příkazového řádku
–all, 243
–help, -h, 243
–version, 243
calibredb-catalog volba příkazového řádku
–fields, 239
–help, -h, 239
–ids, -i, 239
–search, -s, 239
–sort-by, 240
–verbose, -v, 240
–version, 240
calibredb-check_library volba příkazového řádku
–csv, -c, 242
–help, -h, 242
–ignore_extensions, -e, 242
–ignore_names, -n, 242
–report, -r, 242
–version, 242
calibredb-clone volba příkazového řádku
–help, -h, 243
–version, 244
calibredb-custom_columns volba příkazového řádku
–details, -d, 241
–help, -h, 241
–version, 241
calibredb-embed_metadata volba příkazového řádku
–help, -h, 244
–only-formats, -f, 244
–version, 244

calibredb-export volba příkazového řádku
–all, 238
–dont-asciiize, 238
–dont-save-cover, 238
–dont-update-metadata, 238
–dont-write-opf, 238
–formats, 239
–help, -h, 239
–replace-whitespace, 239
–single-dir, 239
–template, 239
–timefmt, 239
–to-dir, 239
–to-lowercase, 239
–version, 239
calibredb-list volba příkazového řádku
–ascending, 235
–fields, -f, 235
–for-machine, 235
–help, -h, 235
–limit, 235
–line-width, -w, 235
–prefix, 235
–search, -s, 235
–separator, 235
–sort-by, 235
–version, 235
calibredb-list_categories volba příkazového řádku
–categories, -r, 242
–csv, -c, 243
–help, -h, 243
–item_count, -i, 243
–quote, -q, 243
–separator, -s, 243
–version, 243
–width, -w, 243
calibredb-remove volba příkazového řádku
–help, -h, 236
–version, 237
calibredb-remove_custom_column volba příkazového řádku
–force, -f, 241
–help, -h, 241
–version, 241
calibredb-remove_format volba příkazového řádku
–help, -h, 237
–version, 237
calibredb-restore_database volba příkazového řádku
–help, -h, 242
–really-do-it, -r, 242
–version, 242
calibredb-saved_searches volba příkazového řádku
–help, -h, 240
–version, 240
calibredb-search volba příkazového řádku
–help, -h, 244
–limit, -l, 244
–version, 244
calibredb-set_custom volba příkazového řádku
–append, -a, 241
–help, -h, 241
–version, 241
calibredb-set_metadata volba příkazového řádku
–field, -f, 238
–help, -h, 238
–list-fields, -l, 238
–version, 238
calibredb-show_metadata volba příkazového řádku
–as-opf, 237
–help, -h, 237
–version, 237
can_be_disabled (atribut calibre.customize.Plugin), 194
CAN_DO_DEVICE_DB_PLUGBOARD (atribut calibre.devices.interface.DevicePlugin), 203
can_get_multiple_covers (atribut calibre.ebooks.metadata.sources.base.Source), 198
can_handle() (metoda calibre.devices.interface.DevicePlugin), 205
can_handle_windows() (metoda calibre.devices.interface.DevicePlugin), 205
CAN_SET_METADATA (atribut calibre.devices.interface.DevicePlugin), 203
canonicalize_internal_url() (metoda calibre.web.feeds.news.BasicNewsRecipe), 43
capabilities (atribut calibre.ebooks.metadata.sources.base.Source), 198
card_prefix() (metoda calibre.devices.interface.DevicePlugin), 206
CatalogPlugin (třída v calibre.customize), 197
category (atribut calibre.customize.PreferencesPlugin), 215
category_order (atribut calibre.customize.PreferencesPlugin), 215
center_navbar (atribut calibre.web.feeds.news.BasicNewsRecipe), 47
change_font() (v modulu calibre.ebooks.oeb.polish.fonts), 279
changed_signal (atribut calibre.gui2.preferences.ConfigWidgetInterface), 215
clean_downloaded_metadata() (metoda calibre.ebooks.metadata.sources.base.Source), 198
cleanup() (metoda ca-

libre.web.feeds.news.BasicNewsRecipe), 43
CLI (třída v calibre.devices.usbms.cli), 211
cli_main() (metoda calibre.customize.Plugin), 196
cli_options (atribut calibre.customize.CatalogPlugin), 197
clone_browser() (metoda calibre.web.feeds.news.BasicNewsRecipe), 43
close_editor() (metoda calibre.gui2.tweak_book.boss.Boss), 281
commit() (metoda calibre.ebooks.oeb.polish.container.Container), 274
commit() (metoda calibre.gui2.preferences.ConfigWidgetInterface), 215
commit_all_editors_to_container() (metoda calibre.gui2.tweak_book.boss.Boss), 281
commit_item() (metoda calibre.ebooks.oeb.polish.container.Container), 274
common_options (atribut calibre.customize.conversion.InputFormatPlugin), 201
common_options (atribut calibre.customize.conversion.OutputFormatPlugin), 202
compress_news_images (atribut calibre.web.feeds.news.BasicNewsRecipe), 47
compress_news_images_auto_size (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
compress_news_images_max_size (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
config_help_message (atribut calibre.ebooks.metadata.sources.base.Source), 198
config_widget (atribut calibre.customize.PreferencesPlugin), 215
config_widget() (metoda calibre.customize.Plugin), 195
config_widget() (třídní metoda calibre.devices.interface.DevicePlugin), 207
ConfigWidgetBase (třída v calibre.gui2.preferences), 216
ConfigWidgetInterface (třída v calibre.gui2.preferences), 215
Container (třída v calibre.ebooks.oeb.polish.container), 273
contains(), **133**
conversion_options (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
convert() (metoda calibre.customize.conversion.InputFormatPlugin), 201
convert() (metoda calibre.customize.conversion.OutputFormatPlugin), 202
copy_cover_to() (metoda calibre.db.cache.Cache), 268
copy_format_to() (metoda calibre.db.cache.Cache), 268
core_usage (atribut calibre.customize.conversion.InputFormatPlugin), 201
cover() (metoda calibre.db.cache.Cache), 268
cover_margins (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
create_action() (metoda calibre.gui2.tweak_book.plugin.Tool), 280
create_inline_toc() (v modulu calibre.ebooks.oeb.polish.toc), 279
create_menu_action() (metoda calibre.gui2.actions.InterfaceAction), 213
create_widget() (metoda calibre.customize.PreferencesPlugin), 215
CSS, **282**
current_container (atribut calibre.gui2.tweak_book.plugin.Tool), 280
currently_editing (atribut calibre.gui2.tweak_book.boss.Boss), 281
custom_field_keys() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
customization_help() (metoda calibre.customize.Plugin), 195
customize_context_menu() (metoda calibre.customize.ViewerPlugin), 217
customize_ui() (metoda calibre.customize.ViewerPlugin), 217

D

data_for_find_identical_books() (metoda calibre.db.cache.Cache), 268
data_for_has_book() (metoda calibre.db.cache.Cache), 268
debug_managed_device_detection() (metoda calibre.devices.interface.DevicePlugin), 204
decode() (metoda calibre.ebooks.oeb.polish.container.Container), 274
deepcopy() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
default_cover() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
delay (atribut calibre.web.feeds.news.BasicNewsRecipe), 48

delete_books() (metoda calibre.devices.interface.DevicePlugin), 207
delete_custom_book_data() (metoda calibre.db.cache.Cache), 268
description (atribut calibre.customize.Plugin), 194
description (atribut calibre.customize.PreferencesPlugin), 215
description (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
detect_managed_devices() (metoda calibre.devices.interface.DevicePlugin), 204
Device (třída v calibre.devices.usbms.device), 210
DevicePlugin (třída v calibre.devices.interface), 203
dirty() (metoda calibre.ebooks.oeb.polish.container.Container), 274
do_user_config() (metoda calibre.customize.Plugin), 195
dont_add_to (atribut calibre.gui2.actions.InterfaceAction), 213
dont_remove_from (atribut calibre.gui2.actions.InterfaceAction), 213
download() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
download_cover() (metoda calibre.ebooks.metadata.sources.base.Source), 200
drop_event() (metoda calibre.gui2.actions.InterfaceAction), 213

E

ebook-convert volba příkazového řádku
–asciiize, 247
–author-sort, 251
–authors, 251
–base-font-size, 247
–book-producer, 251
–change-justification, 247
–chapter, 250
–chapter-mark, 250
–comments, 252
–cover, 252
–debug-pipeline, -d, 252
–disable-dehyphenate, 249
–disable-delete-blank-paragraphs, 249
–disable-fix-indents, 249
–disable-font-rescaling, 247
–disable-format-scene-breaks, 249
–disable-italicize-common-cases, 249
–disable-markup-chapter-headings, 249
–disable-remove-fake-margins, 250
–disable-renumber-headings, 249
–disable-unwrap-lines, 249
–dont-split-on-page-breaks, 246
–duplicate-links-in-toc, 251
–embed-all-fonts, 247
–embed-font-family, 247
–enable-heuristics, 249
–epub-flatten, 246
–epub-inline-toc, 246
–epub-toc-at-end, 246
–expand-css, 247
–extra-css, 247
–extract-to, 246
–filter-css, 247
–flow-size, 246
–font-size-mapping, 248
–help, -h, 245
–html-unwrap-factor, 249
–input-encoding, 246
–input-profile, 245
–insert-blank-line, 248
–insert-blank-line-size, 248
–insert-metadata, 250
–isbn, 252
–keep-ligatures, 248
–language, 252
–level1-toc, 251
–level2-toc, 251
–level3-toc, 251
–line-height, 248
–linearize-tables, 248
–list-recipes, 245
–margin-bottom, 248
–margin-left, 248
–margin-right, 248
–margin-top, 248
–max-toc-links, 251
–minimum-line-height, 248
–no-chapters-in-toc, 251
–no-default-epub-cover, 246
–no-svg-cover, 246
–output-profile, 245
–page-breaks-before, 250
–prefer-metadata-cover, 250
–preserve-cover-aspect-ratio, 246
–pretty-print, 247
–pubdate, 252
–publisher, 252
–rating, 252
–read-metadata-from-opf, -m, 252
–remove-first-image, 250
–remove-paragraph-spacing, 248
–remove-paragraph-spacing-indent-size, 248
–replace-scene-breaks, 249
–search-replace, 250
–series, 252
–series-index, 252

–smarten-punctuation, 249
–sr1-replace, 250
–sr1-search, 250
–sr2-replace, 250
–sr2-search, 250
–sr3-replace, 250
–sr3-search, 250
–start-reading-at, 251
–subset-embedded-fonts, 249
–tags, 252
–timestamp, 252
–title, 252
–title-sort, 252
–toc-filter, 251
–toc-threshold, 251
–toc-title, 247
–unsmarten-punctuation, 249
–use-auto-toc, 251
–verbose, -v, 252
–version, 246
ebook-edit volba příkazového řádku
–detach, 253
–help, -h, 253
–version, 253
ebook-meta volba příkazového řádku
–author-sort, 253
–authors, -a, 253
–book-producer, -k, 253
–category, 253
–comments, -c, 253
–cover, 253
–date, -d, 254
–from-opf, 254
–get-cover, 254
–help, -h, 254
–identifier, 254
–index, -i, 254
–isbn, 254
–language, -l, 254
–lrf-bookid, 254
–publisher, -p, 254
–rating, -r, 254
–series, -s, 254
–tags, 254
–title, -t, 254
–title-sort, 254
–to-opf, 254
–version, 254
ebook-polish volba příkazového řádku
–compress-images, -i, 255
–cover, -c, 255
–embed-fonts, -e, 255
–help, -h, 255
–jacket, -j, 255
–opf, -o, 255
–remove-jacket, 255
–remove-unused-css, -u, 255
–smarten-punctuation, -p, 255
–subset-fonts, -f, 255
–verbose, 256
–version, 256
ebook-viewer volba příkazového řádku
–continue, 256
–debug-javascript, 256
–detach, 256
–full-screen, -f, 256
–help, -h, 256
–open-at, 256
–raise-window, 256
–version, 256
edit_file() (metoda calibre.gui2.tweak_book.boss.Boss), 281
eject() (metoda calibre.devices.interface.DevicePlugin), 205
embed_metadata() (metoda calibre.db.cache.Cache), 268
encoding (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
exists() (metoda calibre.ebooks.oeb.polish.container.Container), 274
extra_css (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
extract_readable_article() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44

F

fast_field_for() (metoda calibre.db.cache.Cache), 269
feeds (atribut calibre.web.feeds.news.BasicNewsRecipe), 48
fetch-ebook-metadata volba příkazového řádku
–authors, -a, 257
–cover, -c, 257
–help, -h, 257
–isbn, -i, 257
–opf, -o, 257
–timeout, -d, 257
–title, -t, 257
–verbose, -v, 257
–version, 257
field_for() (metoda calibre.db.cache.Cache), 269
field_ids_for() (metoda calibre.db.cache.Cache), 269
file_type (atribut calibre.customize.conversion.OutputFormatPlugin), 202
file_types (atribut calibre.customize.CatalogPlugin), 197
file_types (atribut calibre.customize.conversion.InputFormatPlugin), 200
file_types (atribut calibre.customize.FileTypePlugin), 196

file_types (atribut calibre.customize.MetadataReaderPlugin), 196
file_types (atribut calibre.customize.MetadataWriterPlugin), 197
filename_callback() (metoda calibre.devices.usbms.device.Device), 211
filesize() (metoda calibre.ebooks.oeb.polish.container.Container), 274
FileTypePlugin (třída v calibre.customize), 196
filter_css() (v modulu calibre.ebooks.oeb.polish.css), 279
filter_regexps (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
find_identical_books() (metoda calibre.db.cache.Cache), 269
fix_all_html() (v modulu calibre.ebooks.oeb.polish.pretty), 277
fix_html() (v modulu calibre.ebooks.oeb.polish.pretty), 277
for_viewer (atribut calibre.customize.conversion.InputFormatPlugin), 201
format() (metoda calibre.db.cache.Cache), 269
format_abspath() (metoda calibre.db.cache.Cache), 269
format_field() (metoda calibre.ebooks.metadata.book.base.Metadata), 157
format_hash() (metoda calibre.db.cache.Cache), 269
format_metadata() (metoda calibre.db.cache.Cache), 269
FORMATS (atribut calibre.devices.interface.DevicePlugin), 203
formats() (metoda calibre.db.cache.Cache), 270
free_space() (metoda calibre.devices.interface.DevicePlugin), 206
from_files() (v modulu calibre.ebooks.oeb.polish.toc), 279
from_links() (v modulu calibre.ebooks.oeb.polish.toc), 279
from_xpaths() (v modulu calibre.ebooks.oeb.polish.toc), 279

G

generate_item() (metoda calibre.ebooks.oeb.polish.container.Container), 274
genesis() (metoda calibre.gui2.actions.InterfaceAction), 214
genesis() (metoda calibre.gui2.preferences.ConfigWidgetInterface), 215
get_all_standard_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
get_all_user_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
get_annotations() (metoda calibre.devices.usbms.device.Device), 211
get_article_url() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_author_tokens() (metoda calibre.ebooks.metadata.sources.base.Source), 198
get_book_url() (metoda calibre.ebooks.metadata.sources.base.Source), 199
get_book_url_name() (metoda calibre.ebooks.metadata.sources.base.Source), 199
get_book_urls() (metoda calibre.ebooks.metadata.sources.base.Source), 199
get_browser() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_cached_cover_url() (metoda calibre.ebooks.metadata.sources.base.Source), 199
get_categories() (metoda calibre.db.cache.Cache), 270
get_collections() (metoda calibre.devices.interface.BookList), 210
get_cover_url() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_custom_book_data() (metoda calibre.db.cache.Cache), 270
get_device_information() (metoda calibre.devices.interface.DevicePlugin), 205
get_device_uid() (metoda calibre.devices.interface.DevicePlugin), 208
get_driveinfo() (metoda calibre.devices.interface.DevicePlugin), 206
get_feeds() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_file() (metoda calibre.devices.interface.DevicePlugin), 207
get_file_path_for_processing() (metoda calibre.ebooks.oeb.polish.container.Container), 274
get_id_map() (metoda calibre.db.cache.Cache), 270
get_identifiers() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
get_ids_for_custom_book_data() (metoda calibre.db.cache.Cache), 270
get_images() (metoda calibre.customize.conversion.InputFormatPlugin),

201
get_item_id() (metoda calibre.db.cache.Cache), 270
get_item_ids() (metoda calibre.db.cache.Cache), 270
get_item_name() (metoda calibre.db.cache.Cache), 270
get_masthead_title() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_masthead_url() (metoda calibre.web.feeds.news.BasicNewsRecipe), 44
get_metadata() (metoda calibre.customize.MetadataReaderPlugin), 197
get_metadata() (metoda calibre.db.cache.Cache), 270
get_next_series_num_for() (metoda calibre.db.cache.Cache), 270
get_obfuscated_article() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45
get_option() (metoda calibre.devices.interface.DevicePlugin), 209
get_proxy_metadata() (metoda calibre.db.cache.Cache), 270
get_recommended_folders() (v modulu calibre.ebooks.oeb.polish.replace), 277
get_standard_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
get_title_tokens() (metoda calibre.ebooks.metadata.sources.base.Source), 198
get_usage_count_by_id() (metoda calibre.db.cache.Cache), 270
get_user_blacklisted_devices() (metoda calibre.devices.interface.DevicePlugin), 208
get_user_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
guess_type() (metoda calibre.ebooks.oeb.polish.container.Container), 274
gui (atribut calibre.gui2.tweak_book.plugin.Tool), 280
gui_category (atribut calibre.customize.PreferencesPlugin), 215
gui_configuration_widget() (metoda calibre.customize.conversion.InputFormatPlugin), 201
gui_configuration_widget() (metoda calibre.customize.conversion.OutputFormatPlugin), 202
gui_layout_complete() (metoda calibre.gui2.actions.InterfaceAction), 214
gui_name (atribut calibre.customize.PreferencesPlugin), 215
guide_type_map (atribut calibre.ebooks.oeb.polish.container.Container), 274

H

has_book() (metoda calibre.db.cache.Cache), 270
has_format() (metoda calibre.db.cache.Cache), 270
has_html_comments (atribut calibre.ebooks.metadata.sources.base.Source), 198
has_id() (metoda calibre.db.cache.Cache), 271
has_name() (metoda calibre.ebooks.oeb.polish.container.Container), 274
href_to_name() (metoda calibre.ebooks.oeb.polish.container.Container), 274
HTML, **282**

I

icon (atribut calibre.customize.PreferencesPlugin), 215
icon (atribut calibre.devices.interface.DevicePlugin), 203
identify() (metoda calibre.ebooks.metadata.sources.base.Source), 199
identify_results_keygen() (metoda calibre.ebooks.metadata.sources.base.Source), 199
ignore_connected_device() (metoda calibre.devices.interface.DevicePlugin), 208
ignore_duplicate_articles (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
image_url_processor() (třídní metoda calibre.web.feeds.news.BasicNewsRecipe), 45
index_to_soup() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45
init() (metoda calibre.db.cache.Cache), 271
initialization_complete() (metoda calibre.gui2.actions.InterfaceAction), 214
initialize() (metoda calibre.customize.CatalogPlugin), 197
initialize() (metoda calibre.customize.Plugin), 195
initialize() (metoda calibre.gui2.preferences.ConfigWidgetInterface), 215
InputFormatPlugin (třída v calibre.customize.conversion), 200
insert_into_xml() (metoda calibre.ebooks.oeb.polish.container.Container), 274
InterfaceAction (třída v calibre.gui2.actions), 212

InterfaceActionBase (třída v calibre.customize), 214
InternalMetadataCompareKeyGen (třída v calibre.ebooks.metadata.sources.base), 200
is_configured() (metoda calibre.ebooks.metadata.sources.base.Source), 198
is_dir (atribut calibre.ebooks.oeb.polish.container.Container), 274
is_dynamically_controllable() (metoda calibre.devices.interface.DevicePlugin), 208
is_image_collection (atribut calibre.customize.conversion.InputFormatPlugin), 200
is_link_wanted() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45
is_null() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
is_running() (metoda calibre.devices.interface.DevicePlugin), 209
is_usb_connected() (metoda calibre.devices.interface.DevicePlugin), 204
iterlinks() (metoda calibre.ebooks.oeb.polish.container.Container), 275

J

javascript_login() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45

K

keep_only_tags (atribut calibre.web.feeds.news.BasicNewsRecipe), 49

L

language (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
library_changed() (metoda calibre.gui2.actions.InterfaceAction), 214
load_actual_plugin() (metoda calibre.customize.InterfaceActionBase), 214
load_fonts() (metoda calibre.customize.ViewerPlugin), 216
load_javascript() (metoda calibre.customize.ViewerPlugin), 216
load_resources() (metoda calibre.customize.Plugin), 195
load_resources() (metoda calibre.gui2.actions.InterfaceAction), 214
location_selected() (metoda calibre.gui2.actions.InterfaceAction), 214
LRF, **283**
lrf2lrs volba příkazového řádku
–dont-output-resources, 257
–help, -h, 257
–output, -o, 257
–verbose, 257
–version, 257
lrfviewer volba příkazového řádku
–disable-hyphenation, 258
–help, -h, 258
–profile, 258
–verbose, 258
–version, 258
–visual-debug, 258
–white-background, 258
lrs2lrf volba příkazového řádku
–help, -h, 258
–lrs, 258
–output, -o, 258
–verbose, 258
–version, 258

M

MANAGES_DEVICE_PRESENCE (atribut calibre.devices.interface.DevicePlugin), 204
manifest_has_name() (metoda calibre.ebooks.oeb.polish.container.Container), 275
manifest_id_map (atribut calibre.ebooks.oeb.polish.container.Container), 275
manifest_type_map (atribut calibre.ebooks.oeb.polish.container.Container), 275
mark_as_cover() (v modulu calibre.ebooks.oeb.polish.cover), 278
mark_as_titlepage() (v modulu calibre.ebooks.oeb.polish.cover), 278
masthead_url (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
match_regexps (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
max_articles_per_feed (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
MAX_PATH_LEN (atribut calibre.devices.usbms.device.Device), 211
merge() (v modulu calibre.ebooks.oeb.polish.split), 278
Metadata (třída v calibre.ebooks.metadata.book.base), 155
metadata_for_field() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
MetadataReaderPlugin (třída v calibre.customize), 196

MetadataWriterPlugin (třída v calibre.customize), 197
mi (atribut calibre.ebooks.oeb.polish.container.Container), 275
minimum_calibre_version (atribut calibre.customize.Plugin), 194
multisort() (metoda calibre.db.cache.Cache), 271
multisplit() (v modulu calibre.ebooks.oeb.polish.split), 278

N

name (atribut calibre.customize.Plugin), 194
name (atribut calibre.gui2.actions.InterfaceAction), 212
name (atribut calibre.gui2.tweak_book.plugin.Tool), 280
name(), **133**
name_order (atribut calibre.customize.PreferencesPlugin), 215
name_to_abspath() (metoda calibre.ebooks.oeb.polish.container.Container), 275
name_to_href() (metoda calibre.ebooks.oeb.polish.container.Container), 275
names_that_must_not_be_changed (atribut calibre.ebooks.oeb.polish.container.Container), 275
names_that_must_not_be_removed (atribut calibre.ebooks.oeb.polish.container.Container), 275
names_that_need_not_be_manifested (atribut calibre.ebooks.oeb.polish.container.Container), 275
needs_subscription (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
NEWS_IN_FOLDER (atribut calibre.devices.usbms.device.Device), 211
no_stylesheets (atribut calibre.web.feeds.news.BasicNewsRecipe), 49
normalize_path() (třídní metoda calibre.devices.usbms.driver.USBMS), 212
NUKE_COMMENTS (atribut calibre.devices.interface.DevicePlugin), 203

O

oldest_article (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
on_import (atribut calibre.customize.FileTypePlugin), 196
on_postimport (atribut calibre.customize.FileTypePlugin), 196
on_postprocess (atribut calibre.customize.FileTypePlugin), 196
on_preprocess (atribut calibre.customize.FileTypePlugin), 196
open() (metoda calibre.devices.interface.DevicePlugin), 205
open() (metoda calibre.ebooks.oeb.polish.container.Container), 275
open_book() (metoda calibre.gui2.tweak_book.boss.Boss), 281
OPEN_FEEDBACK_MESSAGE (atribut calibre.devices.interface.DevicePlugin), 203
opf (atribut calibre.ebooks.oeb.polish.container.Container), 275
opf_get_or_create() (metoda calibre.ebooks.oeb.polish.container.Container), 275
opf_version (atribut calibre.ebooks.oeb.polish.container.Container), 275
opf_xpath() (metoda calibre.ebooks.oeb.polish.container.Container), 275
options (atribut calibre.customize.conversion.InputFormatPlugin), 201
options (atribut calibre.customize.conversion.OutputFormatPlugin), 202
options (atribut calibre.ebooks.metadata.sources.base.Source), 198
OSX_MAIN_MEM_VOL_PAT (atribut calibre.devices.usbms.device.Device), 211
output_encoding (atribut calibre.customize.conversion.InputFormatPlugin), 201
OutputFormatPlugin (třída v calibre.customize.conversion), 202

P

předpis, **282**
parse_feeds() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45
parse_index() (metoda calibre.web.feeds.news.BasicNewsRecipe), 45
parsed() (metoda calibre.ebooks.oeb.polish.container.Container), 275
path_sep (atribut calibre.devices.interface.DevicePlugin), 203
Plugin (třída v calibre.customize), 194
populate_article_metadata() (metoda calibre.web.feeds.news.BasicNewsRecipe), 46
popup_type (atribut calibre.gui2.actions.InterfaceAction), 212

post_yank_cleanup() (metoda calibre.devices.interface.DevicePlugin), 205
postimport() (metoda calibre.customize.FileTypePlugin), 196
postprocess_book() (metoda calibre.customize.conversion.InputFormatPlugin), 201
postprocess_book() (metoda calibre.web.feeds.news.BasicNewsRecipe), 46
postprocess_html() (metoda calibre.web.feeds.news.BasicNewsRecipe), 46
pref() (metoda calibre.db.cache.Cache), 271
prefer_results_with_isbn (atribut calibre.ebooks.metadata.sources.base.Source), 198
PreferencesPlugin (třída v calibre.customize), 214
prepare_addable_books() (metoda calibre.devices.interface.DevicePlugin), 208
preprocess_html() (metoda calibre.web.feeds.news.BasicNewsRecipe), 46
preprocess_raw_html() (metoda calibre.web.feeds.news.BasicNewsRecipe), 46
preprocess_regexps (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
pretty_all() (v modulu calibre.ebooks.oeb.polish.pretty), 277
pretty_css() (v modulu calibre.ebooks.oeb.polish.pretty), 277
pretty_html() (v modulu calibre.ebooks.oeb.polish.pretty), 277
pretty_xml() (v modulu calibre.ebooks.oeb.polish.pretty), 277
print_version() (třídní metoda calibre.web.feeds.news.BasicNewsRecipe), 47
priority (atribut calibre.customize.Plugin), 194
priority (atribut calibre.gui2.actions.InterfaceAction), 212
PRODUCT_ID (atribut calibre.devices.interface.DevicePlugin), 203
publication_type (atribut calibre.web.feeds.news.BasicNewsRecipe), 50

R

raw_data() (metoda calibre.ebooks.oeb.polish.container.Container), 275
re:test(), **133**
read_backup() (metoda calibre.db.cache.Cache), 271
recipe_disabled (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
recommendations (atribut calibre.customize.conversion.InputFormatPlugin), 201
recommendations (atribut calibre.customize.conversion.OutputFormatPlugin), 202
recursions (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
refresh_gui() (metoda calibre.gui2.preferences.ConfigWidgetInterface), 216
regexp, **283**
register() (metoda calibre.gui2.preferences.ConfigWidgetBase), 216
register_shortcut() (metoda calibre.gui2.tweak_book.plugin.Tool), 280
relpath() (metoda calibre.ebooks.oeb.polish.container.Container), 276
remove_attributes (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
remove_book() (metoda calibre.devices.interface.BookList), 210
remove_books() (metoda calibre.db.cache.Cache), 271
remove_books_from_metadata() (třídní metoda calibre.devices.interface.DevicePlugin), 207
remove_empty_feeds (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
remove_formats() (metoda calibre.db.cache.Cache), 271
remove_from_spine() (metoda calibre.ebooks.oeb.polish.container.Container), 276
remove_from_xml() (metoda calibre.ebooks.oeb.polish.container.Container), 276
remove_item() (metoda calibre.ebooks.oeb.polish.container.Container), 276
remove_items() (metoda calibre.db.cache.Cache), 271
remove_jacket() (v modulu calibre.ebooks.oeb.polish.jacket), 277
remove_javascript (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
remove_tags (atribut calibre.web.feeds.news.BasicNewsRecipe), 50
remove_tags_after (atribut ca-

libre.web.feeds.news.BasicNewsRecipe), 50
remove_tags_before (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
remove_unused_css() (v modulu calibre.ebooks.oeb.polish.css), 279
rename() (metoda calibre.ebooks.oeb.polish.container.Container), 276
rename_files() (v modulu calibre.ebooks.oeb.polish.replace), 277
rename_items() (metoda calibre.db.cache.Cache), 271
replace() (metoda calibre.ebooks.oeb.polish.container.Container), 276
replace_links() (metoda calibre.ebooks.oeb.polish.container.Container), 276
replace_links() (v modulu calibre.ebooks.oeb.polish.replace), 277
requires_version (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
reset() (metoda calibre.devices.interface.DevicePlugin), 204
resolve_internal_links (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
restart_critical (atribut calibre.gui2.preferences.ConfigWidgetInterface), 215
restore_book() (metoda calibre.db.cache.Cache), 271
restore_defaults() (metoda calibre.gui2.preferences.ConfigWidgetInterface), 215
restore_defaults_desc (atribut calibre.gui2.preferences.ConfigWidgetInterface), 215
restore_original_format() (metoda calibre.db.cache.Cache), 271
reverse_article_order (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
rewind_savepoint() (metoda calibre.gui2.tweak_book.boss.Boss), 282
RSS, **282**
run() (metoda calibre.customize.CatalogPlugin), 197
run() (metoda calibre.customize.FileTypePlugin), 196
run_javascript() (metoda calibre.customize.ViewerPlugin), 217

S

safe_read_lock (atribut calibre.db.cache.Cache), 271
sanitize_callback() (metoda calibre.devices.usbms.device.Device), 211
sanitize_path_components() (metoda calibre.devices.usbms.device.Device), 211
save_book() (metoda calibre.gui2.tweak_book.boss.Boss), 282
save_original_format() (metoda calibre.db.cache.Cache), 272
save_settings() (metoda calibre.customize.Plugin), 195
save_settings() (třídní metoda calibre.devices.interface.DevicePlugin), 207
scale_news_images (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
scale_news_images_to_device (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
search() (metoda calibre.db.cache.Cache), 272
serialize_item() (metoda calibre.ebooks.oeb.polish.container.Container), 276
set_all_user_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
set_conversion_options() (metoda calibre.db.cache.Cache), 272
set_cover() (metoda calibre.db.cache.Cache), 272
set_cover() (v modulu calibre.ebooks.oeb.polish.cover), 278
set_driveinfo_name() (metoda calibre.devices.interface.DevicePlugin), 208
set_field() (metoda calibre.db.cache.Cache), 272
set_identifier() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
set_identifiers() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
set_library_info() (metoda calibre.devices.interface.DevicePlugin), 208
set_metadata() (metoda calibre.customize.MetadataWriterPlugin), 197
set_metadata() (metoda calibre.db.cache.Cache), 272
set_modified() (metoda calibre.gui2.tweak_book.boss.Boss), 282
set_option() (metoda calibre.devices.interface.DevicePlugin), 209
set_plugboards() (metoda calibre.devices.interface.DevicePlugin), 207
set_pref() (metoda calibre.db.cache.Cache), 272
set_progress_reporter() (metoda calibre.devices.interface.DevicePlugin), 205
set_spine() (metoda calibre.ebooks.oeb.polish.container.Container), 276
set_user_blacklisted_devices() (metoda ca-

libre.devices.interface.DevicePlugin), 208
set_user_metadata() (metoda calibre.ebooks.metadata.book.base.Metadata), 157
settings() (třídní metoda calibre.devices.interface.DevicePlugin), 207
show_current_diff() (metoda calibre.gui2.tweak_book.boss.Boss), 282
show_editor() (metoda calibre.gui2.tweak_book.boss.Boss), 282
shutdown() (metoda calibre.devices.interface.DevicePlugin), 208
shutting_down() (metoda calibre.gui2.actions.InterfaceAction), 214
simultaneous_downloads (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
skip_ad_pages() (metoda calibre.web.feeds.news.BasicNewsRecipe), 47
SLOW_DRIVEINFO (atribut calibre.devices.interface.DevicePlugin), 204
smart_update() (metoda calibre.ebooks.metadata.book.base.Metadata), 157
sort_index_by() (metoda calibre.web.feeds.news.BasicNewsRecipe), 47
Source (třída v calibre.ebooks.metadata.sources.base), 198
specialize() (metoda calibre.customize.conversion.InputFormatPlugin), 201
specialize_css_for_output() (metoda calibre.customize.conversion.OutputFormatPlugin), 202
specialize_global_preferences() (metoda calibre.devices.interface.DevicePlugin), 208
spine_items (atribut calibre.ebooks.oeb.polish.container.Container), 276
spine_iter (atribut calibre.ebooks.oeb.polish.container.Container), 276
spine_names (atribut calibre.ebooks.oeb.polish.container.Container), 276
split() (v modulu calibre.ebooks.oeb.polish.split), 278
split_jobs() (metoda calibre.ebooks.metadata.sources.base.Source), 198
standard_field_keys() (metoda calibre.ebooks.metadata.book.base.Metadata), 156
STANDARD_METADATA_FIELDS (v modulu calibre.ebooks.metadata.book.base), 157
start_plugin() (metoda calibre.devices.interface.DevicePlugin), 208
startup() (metoda calibre.devices.interface.DevicePlugin), 208
stop_plugin() (metoda calibre.devices.interface.DevicePlugin), 209
summary_length (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
supported_platforms (atribut calibre.customize.Plugin), 194
supports_collections() (metoda calibre.devices.interface.BookList), 210
supports_gzip_transfer_encoding (atribut calibre.ebooks.metadata.sources.base.Source), 198
supports_restoring_to_defaults (atribut calibre.gui2.preferences.ConfigWidgetInterface), 215
sync_booklists() (metoda calibre.devices.interface.DevicePlugin), 207
sync_preview_to_editor() (metoda calibre.gui2.tweak_book.boss.Boss), 282
synchronize_with_db() (metoda calibre.devices.interface.DevicePlugin), 209

T

tag_to_string() (třídní metoda calibre.web.feeds.news.BasicNewsRecipe), 47
tags_older_than() (metoda calibre.db.cache.Cache), 272
template_css (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
template_to_attribute() (metoda calibre.ebooks.metadata.book.base.Metadata), 157
temporary_file() (metoda calibre.customize.Plugin), 195
test_fields() (metoda calibre.ebooks.metadata.sources.base.Source), 198
THUMBNAIL_COMPRESSION_QUALITY (atribut calibre.devices.interface.DevicePlugin), 203
THUMBNAIL_HEIGHT (atribut calibre.devices.interface.DevicePlugin), 203
timefmt (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
timeout (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
title (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
to_html() (metoda calibre.ebooks.metadata.book.base.Metadata),

157
Tool (třída v calibre.gui2.tweak_book.plugin), 279
toolbar_button_popup_mode (atribut calibre.gui2.tweak_book.plugin.Tool), 280
total_space() (metoda calibre.devices.interface.DevicePlugin), 206
touched_fields (atribut calibre.ebooks.metadata.sources.base.Source), 198
type (atribut calibre.customize.Plugin), 194
type (atribut calibre.customize.ViewerPlugin), 216

U

upload_books() (metoda calibre.devices.interface.DevicePlugin), 206
upload_cover() (metoda calibre.devices.usbms.driver.USBMS), 211
URL, **283**
USBMS (třída v calibre.devices.usbms.driver), 211
use_embedded_content (atribut calibre.web.feeds.news.BasicNewsRecipe), 51
use_javascript_to_login (atribut calibre.web.feeds.news.BasicNewsRecipe), 52
user_categories_for_books() (metoda calibre.db.cache.Cache), 273
user_feedback_after_callback (atribut calibre.devices.interface.DevicePlugin), 204
UserAnnotation (atribut calibre.devices.interface.DevicePlugin), 203

V

VENDOR_ID (atribut calibre.devices.interface.DevicePlugin), 203
version (atribut calibre.customize.Plugin), 194
ViewerPlugin (třída v calibre.customize), 216
VIRTUAL_BOOK_EXTENSION_MESSAGE (atribut calibre.devices.interface.DevicePlugin), 203
VIRTUAL_BOOK_EXTENSIONS (atribut calibre.devices.interface.DevicePlugin), 203
volba příkazového řádku
  –dont-notify-gui, 234
  –library-path, 234

W

WANTS_UPDATED_THUMBNAILS (atribut calibre.devices.interface.DevicePlugin), 203
web2disk volba příkazového řádku
  –base-dir, -d, 259
  –delay, 259
  –dont-download-stylesheets, 259
  –encoding, 259
  –filter-regexp, 259
  –help, -h, 259
  –match-regexp, 259
  –max-files, -n, 259
  –max-recursions, -r, 259
  –timeout, -t, 259
  –verbose, 259
  –version, 259
WINDOWS_CARD_A_MEM (atribut calibre.devices.usbms.device.Device), 211
WINDOWS_CARD_B_MEM (atribut calibre.devices.usbms.device.Device), 211
WINDOWS_MAIN_MEM (atribut calibre.devices.usbms.device.Device), 211
windows_sort_drives() (metoda calibre.devices.usbms.device.Device), 211